An Opportunistic, Fast, and Distributed Subchannel and User-Pairing Algorithm for OFDMA

A. Karthik, *Student Member, IEEE,* and Neelesh B. Mehta, *Senior Member, IEEE*

***Abstract*—Channel-aware assignment of subchannels to users in the downlink of an OFDMA system requires extensive feedback of channel state information (CSI) to the base station. Since bandwidth is scarce, schemes that limit feedback are necessary. We develop a novel, low feedback, distributed splitting-based algorithm called SplitSelect to opportunistically assign each subchannel to its most suitable user. SplitSelect explicitly handles multiple access control aspects associated with CSI feedback, and scales well with the number of users. In it, according to a scheduling criterion, each user locally maintains a scheduling metric for each subchannel. The goal is to select, for each subchannel, the user with the highest scheduling metric. At any time, each user contends for the subchannel for which it has the largest scheduling metric among the unallocated subchannels. A tractable asymptotic analysis of a system with many users is central to SplitSelect's simple design. Extensive simulation results demonstrate the speed with which subchannels and users are paired. The net data throughput, when the time overhead of selection is accounted for, is shown to be substantially better than several schemes proposed in the literature. We also show how fairness and user prioritization can be ensured by suitably defining the scheduling metric.**

***Index Terms*—OFDMA, subcarrier allocation, downlink, fading channels, distributed algorithms, splitting algorithm, multiuser diversity, frequency-domain scheduling, feedback.**

I. Introduction

WITH an ever increasing demand for high data-rate services, orthogonal frequency division multiple access (OFDMA) has emerged as a promising downlink technique in next generation wireless systems such as Long Term Evolution (LTE) [1] and WiMAX. In it, the entire system bandwidth is divided into several orthogonal narrowband subcarriers that are used to transmit data to multiple users. A key benefit of OFDMA is multiuser diversity [2]–[4], which is exploited by assigning subcarriers to users on the basis of their instantaneous channel gains. In practice, subcarriers are grouped into subchannels, which are assigned to users, and channel state information (CSI) of subchannels is fed back. For example, in LTE, the smallest frequency allocation unit consists of 12 subcarriers and has a bandwidth of 180 kHz.

A key challenge in realizing the gains from multi-user diversity in OFDMA is the large amount of CSI that needs to be fed back to the scheduler. With N users and S subchannels, NSQ bits need to be fed back to the BS, where Q is the number of bits used to quantize each subchannel gain. Several approaches to reducing feedback have been proposed in the literature, which can be broadly classified into two categories: (i) *Filtering among subchannels:* Users feed back CSI only for their M $(< S)$ strongest subchannels. This is also referred to as best-M feedback [5], [6]. However, the feedback overhead scales linearly with N. (ii) *Filtering among users:* For a subchannel, only users with higher gains feed back CSI for that subchannel. For example, a user feeds back CSI only for subchannels whose gains exceed a threshold [7]–[12].

In [7], just one-bit feedback per subchannel is used. Interestingly, this scheme is sum-rate optimal when the number of users is asymptotically large. However, several thousands of users are needed before the optimality manifests itself. Moreover, important multiple access aspects such as collisions are not accounted for even though the feedback overhead is variable. Thresholding combined with multiple access contention to feed back CSI over a common feedback channel was considered in [8] for a single subchannel and in [9] for multiple subchannels. In [8], [9], users with subchannel gains above a threshold contend over K contention slots to send feedback. The access probability of a user in any slot and the threshold depend on N and K. It is worth noting that in such single threshold schemes, the assignment of a subchannel to its 'best' user is *not* guaranteed. Here, the notion of best user for a subchannel depends on the scheduling criterion used. For example, defining the best user as the one with the highest subchannel gain maximizes system throughput. In the proportional fair scheduler considered in [13], the best user is the one with the highest ratio of subchannel gain to mean subchannel gain.

In [10], [11], for every subchannel, users are partitioned into groups based on their subchannel gains. Users in groups with better subchannel gains send their feedback earlier, over a common feedback channel. This is repeated for all subchannels. In [14], a hybrid feedback scheme that combines best-M feedback and thresholding was presented. In the general order selection based allocation algorithm (GOSA) [15], a user that successfully contends, transmits over the subchannel with the highest gain among the unallocated ones. However, at most one subchannel is assigned to a user and the contention process is not modeled. The effect of fixed feedback rate and finite coherence time on the net throughput was studied in [12], [16], [17]. Schemes that incorporate fairness have also been studied in [5], [18], [19].

Contributions of the paper: In this paper, we design a novel, distributed algorithm called *SplitSelect* to opportunistically

Paper approved by Y. Fang, the Editor for Wireless Networks of the IEEE Communications Society. Manuscript received April 7, 2011; revised August 11, 2011 and October 17, 2011.

Part of this paper was presented at the IEEE Intl. Conf. Commun. (ICC), 2011.

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science (IISc), Bangalore, India (e-mail: {karthik_ab, nbmehta}@ece.iisc.ernet.in).

Digital Object Identifier 10.1109/TCOMM.2012.12.110204

assign subchannels to users with very limited feedback. The algorithm uses a multiple access-based splitting approach that circumvents the need to feed back the channel gain of all subchannels of all users. In a distributed manner, it ensures for every subchannel that the first user that the BS successfully decodes from is its best user. Notably, the subchannel assignment problem and its multiple access aspects are jointly handled.

Briefly, the time-slotted algorithm works as follows. Based on current channel gains, every user maintains a *scheduling metric* for each subchannel, which measures how suitable the user is for that subchannel. The metric depends on the scheduling criterion used [20]. In every slot, a user contends on the basis of its *competing metric*, which is defined to be its largest scheduling metric among subchannels that are yet to be assigned. Only those users whose competing metrics lie between a lower and an upper threshold contend in every slot. Effectively, every user puts its best foot forward in every slot. At the end of a slot, the BS broadcasts an idle/success/collision outcome, based on which the thresholds are updated. A subchannel gets assigned if and only if a success occurs. The algorithm terminates when all the subchannels get assigned. Thus, the algorithm combines and reaps the benefits of filtering among users and among subchannels. Unlike the thresholding schemes [7]–[12], the subchannels are not assigned in a prespecified order.

An insightful analysis for the asymptotic case, where the number of users is large, plays a central role in the design of SplitSelect and determines the threshold update rules. The unique features of the OFDMA scheduling problem makes this algorithm novel even in the well-studied class of splitting algorithms [21]–[27].

Through extensive simulations, we show that SplitSelect is fast and scales well with the number of users. For example, it requires only 2.22 slots, on average, to allocate a subchannel even in a system consisting of 100 users and 25 subchannels. We show that the net throughput achieved by SplitSelect is greater than several schemes proposed in the literature such as single-threshold based random access [9], contention-based opportunistic feedback (OF) scheme [10], and GOSA [15], and is close to ideal selection even for very limited selection durations. Similar performance differences are observed when proportional fairness and user prioritization are incorporated in SplitSelect and the aforementioned benchmark schemes.

The paper is organized as follows. The system model is set up in Section II. The proposed algorithm is defined in Section III. The mathematical reasoning behind it and insights are presented in Section IV. Selection speed and throughput benchmarking with existing schemes are presented in Section V. In Section VI, we show how different notions of fairness and user priority can be incorporated. We conclude in Section VII. Proofs are given in the Appendix.

II. System Model and Terminology

Consider the downlink of an OFDMA cell with N users. The system bandwidth is divided into S frequency-flat subchannels. The square of the amplitude of the complex baseband response of the j^{th} subchannel of user i is denoted by H_{ij} and shall be referred to as its *subchannel gain*. This model covers Rayleigh (non-line-of-sight), Ricean (line-of-sight), Nakagami, and several other common fading distributions. The subchannel gains H_{ij}, for $1 \leq i \leq N$ and $1 \leq j \leq S$, are assumed to be continuous random variables (RVs) that are independent and identically distributed (i.i.d.), as has also been assumed in [6], [8]–[10], [12], [16]–[18], [26]–[28]. Independence across users is a reasonable assumption since the users themselves are spatially separated. Since path loss and shadowing do not change over the system bandwidth of interest, the subchannel gains of a user are clearly identically distributed. Independence across subchannels is justifiable when the subchannel bandwidth is close to the coherence bandwidth. This assumption along with the assumption that the subchannel gains of different users are identically distributed enables analytical tractability. The latter has also been assumed in several related papers [6], [8]–[10], [12], [16]–[18], [26]–[28].[1]

A user i knows its S subchannel gains H_{ij}, $1 \leq j \leq S$, but not the subchannel gains of the other users. At the beginning, the BS does not know any subchannel gain of any user. Every user is assumed to know the cumulative distribution function (CDF) of the subchannel gain, $F_H(.)$, and the number of users, N, since these change very slowly and can be determined [29].

At the start of a coherence interval, the BS pairs users and subchannels using SplitSelect. We assume time to be slotted during this assignment phase. Thereafter, in the remaining duration of the coherence interval, the BS transmits data on a subchannel to its assigned user. For feeding back CSI, the users share a common multiple access channel to the BS. The BS can decode the CSI fed back when exactly one user transmits. However, when multiple users transmit simultaneously, a collision results and the BS cannot decode any of the transmissions [21], [26], [27]. The collision model helps us focus on the core multiple access aspect of the problem and enables a study of collision resolution [22]. Physically, energy detection methods and link budget provisioning can enable the BS receiver to *correctly* differentiate among the no signal (idle), decodable signal (success), and non-decodable high energy signal (collision) cases.

At the end of every slot, the BS broadcasts an idle, success, or collision outcome indicating whether $0,\ 1$, or multiple users transmitted. Every user is assumed to receive this feedback from the BS without error, which is justifiable given the low feedback payload [21], [26], [27].

We introduce the following important terms, which will be used throughout the paper.

1) Scheduling Metric: For a user i, the *scheduling metric*, S_{ij}, is a real value associated with subchannel j. For a subchannel j, the goal of the selection scheme is to assign the user with the highest scheduling metric, *i.e.*, $\arg\max_i S_{ij}$, to it. As discussed in Section VI, different scheduling metrics imply different notions of fairness and user prioritization.

2) User Metric: When the scheduling metrics are i.i.d., a *user metric* X_{ij} is associated with a scheduling metric S_{ij} as follows:

$$X_{ij} \triangleq N\left(1 - F_S(S_{ij})\right), \tag{1}$$

[1] Note that even without these two assumptions, SplitSelect will assign the best user to each subchannel. We shall revisit these assumptions in Section VI.

where $F_S(\cdot)$ is the CDF of S_{ij}. Now, regardless of the specific distribution $F_S(\cdot)$, X_{ij} is a uniformly distributed RV in $(0, N)$ [30]. Since $F_S(\cdot)$ is monotonically increasing, *the problem of finding the user with the highest scheduling metric is equivalent to finding the user with the smallest user metric.* Since S_{ij} are i.i.d., so are X_{ij}. Thus, all continuous CDFs, $F_S(\cdot)$, are now handled in a single framework.

3) Competing Metric and Competing Subchannel: For a user i in timeslot k, the *competing metric*, $m_i(k)$, is the minimum of the user metrics of the subchannels that are yet to be allocated:

$$m_i(k) = \min_j \{X_{ij} : \text{Subchannel } j \text{ has not been allocated before slot } k\}. \quad (2)$$

The corresponding subchannel at which the above minimum occurs is called the *competing subchannel* of user i in slot k. Note that a user's competing metric can change only at the end of a timeslot in which a subchannel is allocated.

III. User-to-Subchannel Pairing Algorithm: Definition

For clarity, we first formally define in this section the terminology, steps, and the state variables of the algorithm. A systematic mathematical development and explanation of the algorithm follows in Section IV.

The algorithm allocates subchannels successively. However, the order of subchannel allocation depends on the specific realization of the various user metrics. We shall call the time duration required to allocate a subchannel a *subchannel allocation period* (SAP). After every allocation, a new SAP starts. Fig. 1 illustrates an example of the subchannel assignment process, which is formally defined below. During the first SAP, the competing subchannels of users U_1 and U_2 are S_3 and S_1, respectively, and S_3 gets assigned to U_1 as its subchannel gain is the highest. In the second SAP, U_1 changes its competing subchannel to its next best subchannel, S_2, while U_2's competing subchannel is still S_1. After contention, S_1 gets assigned to U_2. During the third SAP, the competing subchannel of both the users is S_2, which gets assigned to U_1. Thus, all three subchannels have been assigned to their respective best users.

State variables: The algorithm maintains six state variables in every slot k: L_k, R_k, V_k, Δ_k, C_k, and ρ_k.[2] The variables L_k and R_k are called the *left* and *right* thresholds, respectively. A user i transmits in timeslot k only if its competing metric lies between L_k and R_k:

$$L_k < m_i(k) < R_k. \quad (3)$$

When a user transmits, it sends its identity, its competing subchannel, and the gain of its competing subchannel (or a Q-bit quantized version thereof). This requires $\lceil \log_2(NS) + Q \rceil$ bits, where $\lceil \cdot \rceil$ denotes the ceil function.

The variable C_k is a two-state variable that takes values 1 or 0, which respectively indicate whether a collision has occurred

[2]Every user stores and computes these variables independently based only on the outcome broadcast by the BS. The values of the state variables will turn out to be the same across all users in every timeslot.

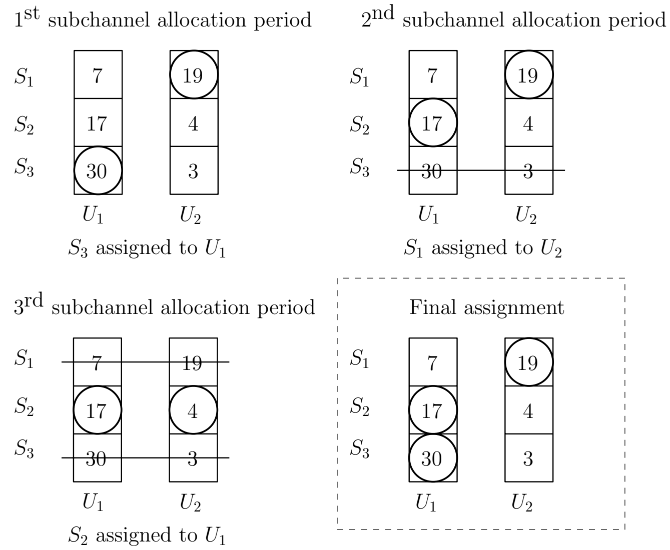


Fig. 1. Illustration of subchannel assignment for $N = 2$ users (U_1 and U_2) and $S = 3$ subchannels (S_1, S_2, and S_3). The number within each box represents the corresponding subchannel gain of the user. For example, the gain of subchannel S_2 of user U_1 is 17. A circled box represents the competing subchannel of the user during a subchannel allocation period (SAP).

or not during the current subchannel allocation period. The algorithm is said to be in *collision mode* if $C_k = 1$, and is in *non-collision mode* otherwise. The variable ρ_k denotes the total number of unallocated subchannels at the start of the k^{th} slot. As elaborated below, the interval $(V_k, V_k + \Delta_k)$ is the sub-interval of (L_k, R_k) in which the probability of finding at least one competing metric from among the N users is significant.

Initialization ($k = 1$): At the start of the first slot, the state variables are initialized as follows: $L_1 = 0$, $R_1 = p_e/S$, $V_1 = 0$, $\Delta_1 = p_e/S$, $C_1 = 0$, and $\rho_1 = S$. Here, p_e is called the *idle contention load parameter*. It controls the average number of users that transmit in a slot, and shall be optimized later.

Outcomes: At the end of each slot, the BS broadcasts one of the following three outcomes: (i) *idle*, if no user transmitted, (ii) *collision*, if at least 2 users transmitted and collided, or (iii) *success*, if one user transmitted and was, thus, decoded by the BS. This user is called the *winner* user of that slot, and its competing subchannel is assigned by the BS to it. The BS then broadcasts this subchannel's number to all the users along with the success outcome. Conservatively, the BS broadcast overhead is $\lceil \log_2(3 + S) \rceil$ bits.

State variable update rules: The state variables are updated as follows.

- *If slot k is an idle:* Δ_{k+1} is updated as follows:

$$\Delta_{k+1} = \begin{cases} p_e/\rho_k, & C_k = 0 \\ \Delta_k/2, & C_k = 1 \end{cases}. \quad (4)$$

The remaining state variables are updated as

$$V_{k+1} = L_{k+1} = R_k, C_{k+1} = C_k, \text{and } \rho_{k+1} = \rho_k. \quad (5)$$

- *If slot k is a success:* Δ_{k+1} is updated as follows:

$$\Delta_{k+1} = \begin{cases} p_e/(\rho_k - 1), & C_k = 0 \\ p_e \Delta_k/\eta_k, & C_k = 1, \ \eta_k > p_e \\ \Delta_k + \frac{p_e - \eta_k}{\rho_k - 1}, & C_k = 1, \ \eta_k \le p_e \end{cases}, \quad (6)$$

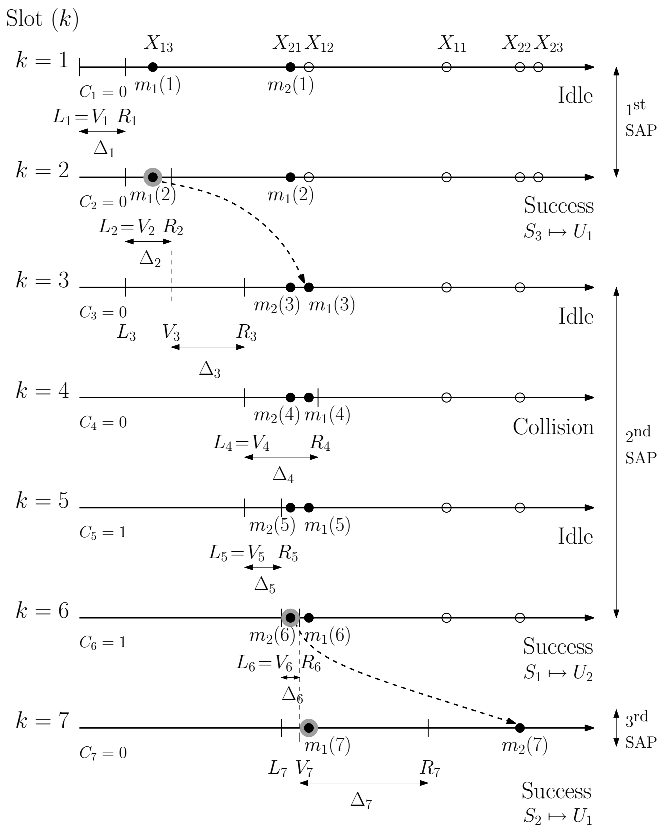


Fig. 2. Timeline of the assignment process for the example shown in Figure 1. All metrics of the two users are shown as circles and are ordered on the real line. A darkened circle represents the competing metric of a user in a timeslot. A dotted arrow shows the updating of the competing metric of a winner user.

where $\eta_k \triangleq \left(1-\frac{1}{\rho_k}\right)\left(1+\frac{\Delta_k\rho_k}{2+\Delta_k\rho_k}\right)$. The remaining state variables are updated as

$$V_{k+1}=R_k, L_{k+1}=L_k, C_{k+1}=0, \text{and } \rho_{k+1}=\rho_k-1. \tag{7}$$

- *If slot k is a collision:* In this case,

$$\Delta_{k+1}=\Delta_k/2, \quad \text{and} \tag{8}$$

$$V_{k+1}=V_k, L_{k+1}=L_k, C_{k+1}=1, \text{ and } \rho_{k+1}=\rho_k. \tag{9}$$

The right threshold is always set as: $R_{k+1}=V_{k+1}+\Delta_{k+1}$.[3]

Termination: The algorithm terminates when ρ_k reaches 0.

Fig. 2 illustrates how the state variables can get updated over time for the same example that is shown in Figure 1.

IV. Design and Analysis of the Algorithm

We now explain the mathematical reasoning behind the various update rules of the algorithm and the roles of its state variables. For brevity, we shall say that a user lies in an interval when its competing metric lies in the interval. We say that a user metric lies to the left or right of a real number x when it is lesser than or greater than x, respectively. The probability of an event A is denoted by $\mathbf{P}[A]$. For an RV X, its expectation and CDF are denoted by $\mathbf{E}[X]$ and $F_X(.)$, respectively. And, $A_N \overset{N\to\infty}{=} B$ shall denote $\lim_{N\to\infty} A_N = B$.

The update rules of the algorithm are based on the following two important design principles: (i) In the non-collision mode, p_e users transmit in the next slot, on average. (ii) In the collision mode, half the colliding users, on average, transmit in the next slot. While these two ideas are common to several splitting algorithms [21], [26], [27], the manner in which they manifest themselves in our algorithm is quite different. The update rules are designed by analyzing a system with a large number of users ($N\to\infty$), for which analytically tractable closed-form expressions can be derived. As shown in Section V, the asymptotic analysis is accurate for N as small as 10.

A. Updating of C_k and ρ_k

The algorithm tracks using the variable C_k whether a collision has occurred or not during the current SAP. When it occurs, C_k is set to 1. It is reset to 0 only after a success slot, at which point the SAP and collision mode end. C_k does not change its value otherwise. Clearly, $C_1=0$.

Recall that ρ_k denotes the number of subchannels that are yet to be assigned at the start of slot k. Since a subchannel gets assigned only in a success slot, ρ_k is decremented by one after and only after a success slot. Clearly, $\rho_1=S$.

B. Significance of Δ_k and V_k

We now describe the variables Δ_k and V_k, which are key to understanding the algorithm.

Let i denote the winner user in a success slot k. Then, we know that $m_i(k)\in(L_k,R_k)$. From (2), $m_i(k+1)>m_i(k)$. Further, since $m_i(k)$ can lie anywhere in the interval (L_k,R_k), $m_i(k+1)$ can also lie anywhere to the right of L_k. Therefore, *the left threshold must remain unchanged after a success slot,* as done in (7), and only R_k increases to R_{k+1}.

However, the competing metrics of the users in the next slot $(k+1)$ are *no longer* uniformly distributed over the transmission interval (L_{k+1},R_{k+1}) as the following result shows, for large N.

Proposition 1: Let k be a success slot. The probability of finding the competing metric of *any* user in (L_k,R_k) in slot $k+1$ tends to 0 for large N.

Proof: The proof is given in Appendix A. ■

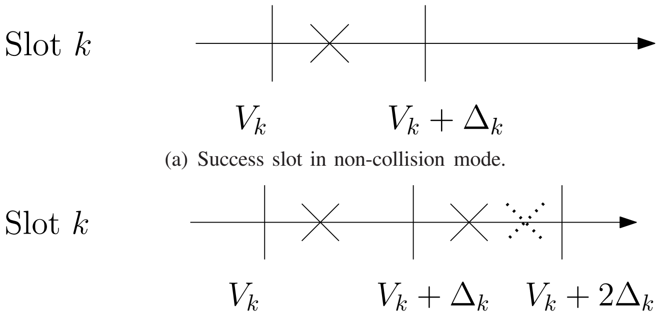
Thus, only users that lie in (R_k,R_{k+1}) will contend in slot $k+1$ after a success slot k for large N.

It is to track the sub-interval from which the users will contend that we introduce V_k and Δ_k, with $R_k=V_k+\Delta_k$. The interval (V_k,R_k) is, by construction, the sub-interval of (L_k,R_k) in which the probability of finding a competing metric is non-zero for large N. And, Δ_k is the length of the sub-interval. Initially ($k=1$), the competing metrics can lie anywhere in $(0,N)$. Hence, the algorithm starts with $L_1=V_1=0$.

C. Idle Slot k

When an idle occurs in slot k, it implies that all users must lie to the right of R_k. Therefore, $L_{k+1}=R_k$ and $V_{k+1}=R_k$, as given in (5).

[3] Even though R_k is entirely determined by V_k and Δ_k, we include it as a state variable for ease of description.



(a) Success slot in non-collision mode.

(b) Success slot in collision mode.

Fig. 3. Location of competing metrics in a success slot.

1) Idle Slot k in Non-collision Mode: The following result shows how to update Δ_{k+1}.

Proposition 2: Let slot k be an idle slot in non-collision mode ($C_k = 0$). For large N, the expected number of users that lie in the interval (R_k, R_{k+1}) at the end of slot k equals $\Delta_{k+1}\rho_k$.

Proof: The proof is given in Appendix B. ■

To ensure that an average of p_e users transmit in slot $k+1$, we must, therefore, equate $\Delta_{k+1}\rho_k$ to p_e. This gives $\Delta_{k+1} = \frac{p_e}{\rho_k}$, as in (4). Thus, $\Delta_1 = \frac{p_e}{\rho_1} = \frac{p_e}{S}$.

2) Idle Slot k in Collision Mode: This implies that the most recent collision must have occurred in slot $k-1$ or earlier. Let it have occurred in slot l. Therefore, the interval (V_l, R_l) must contain at least two users. Thereafter, slots $l+1, \ldots, k$ are all idle. From (8), $\Delta_{l+1} = \Delta_l/2$. Hence, the length of (R_{l+1}, R_l) is Δ_{l+1}. Since slot $l+1$ is idle, the colliding users must lie in the interval (R_{l+1}, R_l). As per the second design principle enunciated in the beginning of this section, (R_{l+1}, R_l) is halved in the next slot, *i.e.*, $\Delta_{l+2} = \Delta_{l+1}/2$. Proceeding likewise, we see that $\Delta_{k+1} = \Delta_k/2$.

D. Success Slot k

When k is a success slot, as discussed earlier in Section IV-B, the left threshold of the transmission interval should not change, *i.e.*, $L_{k+1} = L_k$. Moreover, from Proposition 1, the probability of finding any user to the left of R_k is zero for large N. Hence, $V_{k+1} = R_k$, as given in (7).

We now analyze the cases where the success occurs in the collision or non-collision modes.

1) Success Slot k in Non-collision Mode: In this case, we know that the winner user in slot k lies in the interval $(V_k, V_k + \Delta_k)$. And, all the other users lie to the right of $V_k + \Delta_k$, as illustrated in Fig. 3(a). In this case, the following result holds.

Proposition 3: For large N, the expected number of users in $(R_k, R_k + \Delta_{k+1})$ in slot $k+1$, given that slot k is a success slot in non-collision mode, equals $(\rho_k - 1)\Delta_{k+1}$.

Proof: The proof is given in Appendix C. ■

Thus, $\Delta_{k+1} = \frac{p_e}{\rho_k - 1}$ (as per (6)) ensures that p_e users transmit, on average, in slot $k+1$.

2) Success Slot k in Collision Mode: In this case, the winner user in slot k lies in the interval $(V_k, V_k + \Delta_k)$ and at least one other user lies in the interval $(V_k + \Delta_k, V_k + 2\Delta_k) = (R_k, R_k + \Delta_k)$. This is because a collision has occurred before slot k and the system is still in the collision mode at the beginning of slot k. This scenario is illustrated in Fig. 3(b).

In general, 2 to $N-1$ users can lie in $(R_k, R_k + \Delta_k)$. For $p_e \in [1, 2]$, the odds that one or two users lie in the interval $(R_k, R_k + \Delta_k)$ are considerably more than three or more users lying in it. Large values of p_e (> 2) lead to too many collisions, which slows down the algorithm. On the other hand, for $p_e < 1$, a large number of slots get wasted as idle slots. Hence, these values are not of interest [21], [27]. We, therefore, focus on the two cases of one or two users lying in $(R_k, R_k + \Delta_k)$. For each case, we derive its probability of the occurrence and the average number of users that will lie in the interval $(R_k, R_k + \Delta_{k+1})$, and eventually determine Δ_{k+1}. Let $Z_{k+1}(\Delta_{k+1})$ denote the number of users that lie in $(R_k, R_k + \Delta_{k+1})$ in slot $k+1$ when k is a success slot in collision mode.

Proposition 4: The average number of users that lie in $(R_k, R_k + \Delta_{k+1})$ in slot $k+1$ when exactly one user lies in $(R_k, R_k + \Delta_k)$ in slot success slot k is given as follows for large N:

$$\mathbf{E}[Z_{k+1}(\Delta_{k+1}) |\, 1 \text{ user in } (R_k, R_k + \Delta_k) \text{ in success slot } k] \overset{N\to\infty}{=} \begin{cases} \left(1 - \frac{1}{\rho_k}\right) \\ \quad +(\rho_k - 1)(\Delta_{k+1} - \Delta_k), & \Delta_{k+1} > \Delta_k \\ \left(1 - \frac{1}{\rho_k}\right)\frac{\Delta_{k+1}}{\Delta_k}, & \Delta_{k+1} \in [0, \Delta_k] \end{cases}. \tag{10}$$

The probability, p_1, of finding exactly one user in $(R_k, R_k + \Delta_k)$ in a success slot k given that at most two users lie in $(R_k, R_k + \Delta_k)$ is $p_1 \overset{N\to\infty}{=} \frac{2}{2 + \Delta_k\rho_k}$.

Proof: The proof is given in Appendix D. ■

Proposition 5: The average number of users that lie in $(R_k, R_k + \Delta_{k+1})$ in slot $k+1$ when exactly two users lie in $(R_k, R_k + \Delta_k)$ in success slot k is given as follows for large N:

$$\mathbf{E}[Z_{k+1}(\Delta_{k+1}) |\, 2 \text{ users in } (R_k, R_k + \Delta_k) \text{ in success slot } k] \overset{N\to\infty}{=} \begin{cases} 2\left(1 - \frac{1}{\rho_k}\right) \\ \quad +(\rho_k - 1)(\Delta_{k+1} - \Delta_k), & \Delta_{k+1} > \Delta_k \\ 2\left(1 - \frac{1}{\rho_k}\right)\frac{\Delta_{k+1}}{\Delta_k}, & \Delta_{k+1} \in [0, \Delta_k] \end{cases}. \tag{11}$$

The probability, p_2, of finding exactly two users in $(R_k, R_k + \Delta_k)$ in success slot k given that there are at most two users in $(R_k, R_k + \Delta_k)$ is $p_2 \overset{N\to\infty}{=} \frac{\Delta_k\rho_k}{2 + \Delta_k\rho_k}$.

Proof: The proof is given in Appendix E. ■

Hence, from the law of total expectation, the expected number of users in the interval $(R_k, R_k + \Delta_{k+1})$ in slot $k+1$ given that k was a success slot in collision mode equals

$$\mathbf{E}[Z_{k+1}(\Delta_{k+1}) | k \text{ is a success slot in collision mode}] \overset{N\to\infty}{=} \begin{cases} (p_1 + 2p_2)\left(1 - \frac{1}{\rho_k}\right) \\ \quad +(\rho_k - 1)(\Delta_{k+1} - \Delta_k), & \Delta_{k+1} > \Delta_k \\ (p_1 + 2p_2)\left(1 - \frac{1}{\rho_k}\right)\frac{\Delta_{k+1}}{\Delta_k}, & \Delta_{k+1} \in [0, \Delta_k] \end{cases}.$$

Equating the above expression to p_e and simplifying yields the update rule in (6).

E. Collision Slot k

Since k is a collision slot, there are at least two colliding users in the interval (L_k, R_k). From Proposition 1, all these colliding users must lie with probability one in (V_k, R_k), for large N. Thus, using the second design principle stated in the beginning of this section, the left half of the interval (V_k, R_k) becomes the new contention interval. Hence, $\Delta_{k+1} = \Delta_k/2$, as given in (8), and $V_{k+1} = V_k$, as given in (9). As discussed in Section IV-B, $L_{k+1} = L_k$.

We have, thus, explained all the update rules of the algorithm.

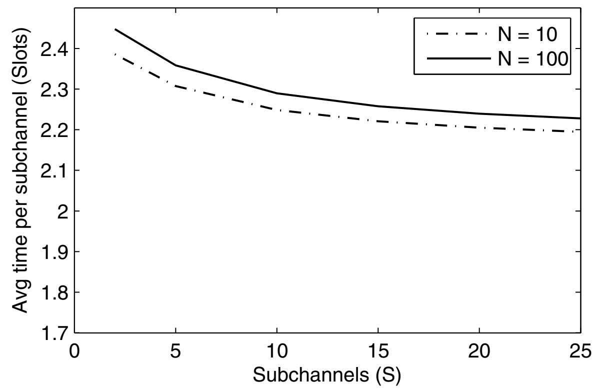


Fig. 4. Zoomed-in view of the average number of timeslots required to assign a subchannel as a function of the number of subchannels, S, for different N.

V. Speed of Selection and Throughput Benchmarking

We now study in detail the performance of the proposed algorithm using Monte Carlo simulations that use 10^6 channel realizations. The different subchannel gains are taken to be exponentially distributed with unit mean. We set $p_e = 1.1$, since this value minimizes the average time required to assign a subchannel for various values of N and S. Values of p_e that exceed 1.1 lead to many collisions, which slows down the algorithm. On the other hand, for $p_e < 1.1$, many slots get wasted as idle slots. For example, for 10 users with $p_e = 2$, the probability of a collision outcome is 62%, where as, for $p_e = 0.5$, the probability of an idle outcome is 60%.

First, we study how fast and scalable the algorithm is in Section V-A. Thereafter, we benchmark its performance with several other schemes proposed in the literature.

A. Average Time Taken to Assign Subchannels

Fig. 4 plots the average number of slots per subchannel required by the algorithm to assign *every* subchannel to its best user as a function of the number of subchannels, S. To allocate 25 subchannels, the algorithm requires just 2.18 and 2.22 slots per subchannel for $N = 10$ and 100 users, respectively. Further, the variation in the average time when N changes from 10 to 100 is just 1.4%. Thus, the algorithm is fast and scales well with N. Observe that the average selection time per subchannel decreases as S increases. This is because the algorithm does not assign subchannels in a pre-defined order. Instead, it lets different users compete for different subchannels simultaneously.

B. Benchmark Schemes

We now briefly describe the schemes against which we compare the performance of SplitSelect.

1) Contention-based OF [10]: For every subchannel, the users are partitioned into equi-probable groups based on their subchannel gains. Each group is associated with a feedback slot. In each slot, users in the corresponding group send their feedback over a common contention-based channel. User groups with better subchannel gains access feedback slots earlier.

2) Single-threshold Based Random Access [9]: For every subchannel, users whose gains exceed a threshold contend to send feedback over K slots with an access probability of p in each slot. After K slots, one of the users that successfully transmitted in any of the K slots is randomly assigned to the subchannel. If no slot led to a success, then one among the N users is picked randomly. Note that this scheme is a practical implementation of the asymptotically optimal 1-bit threshold feedback scheme [7]. In all plots, we numerically optimize both the threshold and p for each set of system parameters in order to provide a fair comparison.

3) GOSA [15]: A user that succeeds in a contention process is assigned the best subchannel among the unallocated ones. A user is assigned atmost one subchannel. Since the contention process is not modeled in [15], we show idealized results for GOSA that do not account for any contention overhead. As per [15], in the contention process, a contending user is selected with uniform probability.

C. Throughput Benchmarking

To illustrate the consequences of such fast user-subchannel pairing and to enable comparisons with two different assignment phase duration models considered in the literature, we consider the following two models:

1) *Fixed assignment phase duration:* In this setup, the total time to assign all the S subchannels is limited to SB slots, *i.e.*, B slots per subchannel [9], [10], [12]. Unallocated subchannels, if any, at the end of the selection phase are assigned randomly among the users in order to utilize every subchannel for data transmission.
2) *Variable assignment phase duration:* Here, the time required to assign all the S subchannels is variable and depends on the algorithm, and data transmission on all subchannels occurs thereafter for D slots [21], [26].

For both these models, we benchmark the performance of SplitSelect against the schemes discussed in Section V-B.

For the throughput comparisons, we set the scheduling metric as $S_{ij} = H_{ij}$. Thus, for a subchannel j, the user with the *highest subchannel gain* is assigned to it. For all the algorithms, the throughput of subchannel j when it is assigned to user i is calculated, for the purpose of illustration, using

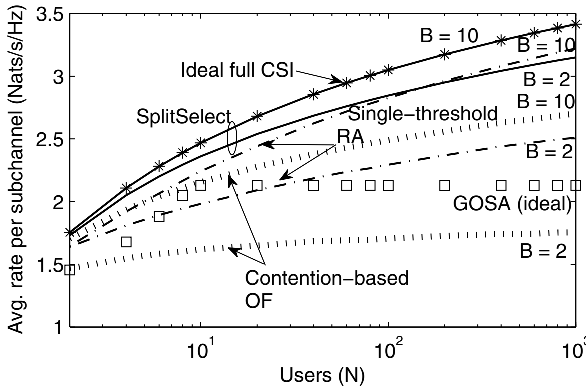


Fig. 5. Comparison of the average rate per subchannel at 6 dB SNR: (i) SplitSelect, (ii) Contention-based OF (the algorithm's parameters are $s = 0.9$ and $c_a = 3$ as per [10, Section IV]), (iii) Single-threshold based random access (RA), and (iv) Idealized general order selection based allocation algorithm (GOSA). The number of slots available for selection per subchannel is B ($S = 10$).

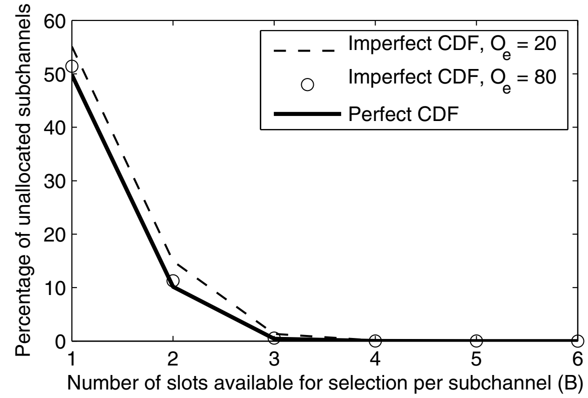


Fig. 6. Percentage of unallocated subchannels at the end of SB slots with perfect and imperfect CDF knowledge ($S = 10$, $N = 10$). In the case of imperfect CDF, the CDF is estimated from O_e observations.

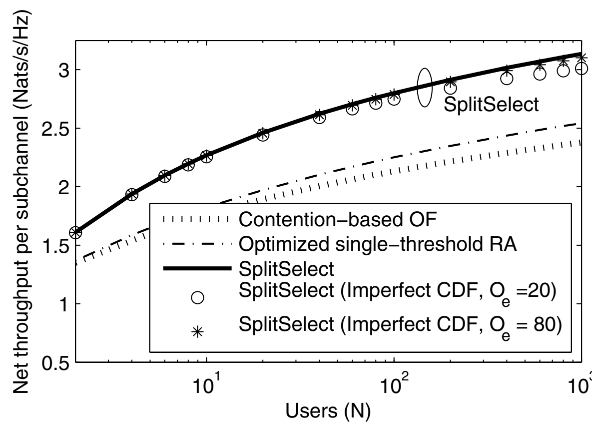


Fig. 7. Comparison of net throughput per subchannel for $D = 250$, SNR $\gamma = 6$ dB, and $S = 10$ of: (i) SplitSelect (perfect and imperfect CDF), (ii) Contention-based OF (the algorithm's parameters are $s = 0.9$ and $c_a = 3$ as per [10, Section IV]), and (iii) Single-threshold based random access (RA) ($K = 5$). In the case of imperfect CDF, the CDF is estimated from O_e observations. Unless otherwise specified, perfect CDF is considered for the schemes.

the Shannon formula: $\log_e(1 + \gamma H_{ij})$ nats/s/Hz, where γ is the signal-to-noise ratio (SNR).[4] Note that the methodology applies equally well to any other mapping between the subchannel gain and throughput.

1) Fixed Assignment Phase Duration: Since the selection duration is fixed, we can directly compare the average throughputs achieved by the algorithms over all subchannels in the data-transmission phase. These are plotted in Fig. 5 for $B = 2$ (limited time for assignment) and $B = 10$. For both values of B, SplitSelect achieves substantially higher throughputs. For example, for $B = 2$, the throughput gains over single-threshold based random access are 19% and 26% for $N = 10$ and 1000 users, respectively. The corresponding gains over contention-based OF are 44% and 80%, and over idealized GOSA are 10% and 48%. The contention-based OF scheme performs poorly primarily because collisions are not resolved, which increases the odds of a sub-optimal subchannel assignment. Also, notice that the throughput of idealized GOSA saturates as N increases, despite its contention overhead not being accounted for. Fig. 5 also plots the throughput achieved by an ideal system with full CSI at the BS. SplitSelect achieves the throughput of the ideal system when $B = 10$, unlike the benchmark schemes.

By its very design, SplitSelect assigns subchannels with higher gains earlier. Thus, not only do more subchannels get assigned within SB slots, the ones that are unassigned are the 'weaker' ones for which random user assignment incurs the least performance loss. This contributes to the substantial gains observed. We delve into this aspect deeper in Fig. 6, which plots the percentage of subchannels that remain unassigned as a function of B for $N = 10$ users. We see that for $B = 2$, only 10.2% of the subchannels are unassigned when users have perfect CDF knowledge. For $B \geq 4$, there are almost no unallocated subchannels.

[4]Note that the use of the Shannon formula overestimates the throughput achieved by the single-threshold random access scheme because the BS does not know the subchannel gain, but only that it exceeds the threshold. In [9], the BS, therefore, polls each selected user to determine its channel gain. For contention-based OF and SplitSelect, given the many thresholds they use, the overestimation is negligible even with $Q = 0$ bits of CSI feedback.

2) Variable Assignment Phase Duration: In this case, we compare the net throughput per subchannel which is given by $\overline{R}\frac{D}{\overline{T}_s+D}$, where $\overline{T}_s$ is the average duration of the assignment phase and $\overline{R}$ is the average rate per subchannel when data transmission occurs. Fig. 7 compares the net throughput of all the schemes except GOSA. The performance of idealized GOSA is not shown since its selection overhead, which affects the net throughput, is not modeled in [15]. For single-threshold based random access, K is set to 5 since it yielded the highest net throughput. The net throughput of SplitSelect is 24% more than single-threshold based random access and 30% more than contention-based OF for $N = 100$ users due to its faster subchannel assignment.

3) Impact of Imperfect CDF Knowledge: Figs. 6 and 7 also show the impact of imperfect CDF knowledge on SplitSelect. The imperfect CDF is generated as per the parametric estimation model of [29]. As before, the subchannel power gains are unit mean i.i.d. exponential RVs. However, the system does not know the mean and the CDF, and must estimate them from O_e observations of the subchannel power gains. This estimated CDF is then used by SplitSelect. The results are averaged over 10^5 different sets of O_e observations. With just 20 observations, we observe that the percentage of unallocated subchannels in Fig. 6 and the net throughput in Fig. 7 is close to the scenario with perfect CDF knowledge. With 80 observations, the performance of SplitSelect in both figures is as good as that with perfect CDF knowledge. For every value of N, SplitSelect with imperfect CDF outperforms the benchmark schemes, despite them having access to perfect CDF knowledge.

VI. Incorporating Fairness and User Priority

We now show how different notions of fairness and even user prioritization can be accommodated in SplitSelect by suitably redefining the scheduling metric. Its performance is also compared with the benchmark schemes under the same fairness and prioritization criteria.

A. Fairness

Proportional fairness (PF) can be included by defining the scheduling metric as $S_{ij} = \frac{R_{ij}}{E[R_i]}$, where R_{ij} is the instantaneous rate of user i over subchannel j and $E[R_i]$ is the average throughput of user i over all subchannels [3], [31], [32].[5] An alternate PF scheduler that has been extensively used in the literature [13], [28], [33] selects the user with the highest ratio of instantaneous channel power gain to the average channel power gain. It can be implemented as $S_{ij} = \frac{H_{ij}}{E[H_{ij}]}$. SplitSelect can also implement a different CDF-based notion of time-fairness proposed in [34] by using $S_{ij} = F_{H_{ij}}(H_{ij})$, where $F_{H_{ij}}$ is the CDF of H_{ij}. Similarly, max-min fairness [35] and fairness with minimum-rate guarantees [5] can also be handled.

Comparisons: We consider a system with $N = 15$ users and $S = 6$ subchannels. The users' subchannel gains are not identically distributed, so that the effect of ensuring fairness can be clearly seen. This is modeled by setting the mean subchannel gain of user i to be α^{i-1}, where $\alpha \geq 1$ and $1 \leq i \leq N$. The scheduling metric of user i over subchannel j is $S_{ij} = \frac{H_{ij}}{E[H_{ij}]} = \frac{H_{ij}}{\alpha^{i-1}}$ [13] for all the schemes. A user i computes X_{ij} according to (1).

Fig. 8 plots the average throughput per subchannel for each user for SplitSelect when $B = 2$. Also plotted are the corresponding values for contention-based OF, single-threshold based random access, and the reference case of ideal selection. We observe that all the schemes exploit multi-user diversity and ensure larger throughputs to users with larger mean subchannel gains, which is a hallmark of the PF scheduler. All the schemes were observed to be indeed time-fair across users (figure not shown). However, the throughput of SplitSelect is better than all the benchmark schemes for every user. Further, its performance is close to ideal selection even for $B = 2$. This is not so for the benchmark schemes. Similar trends hold for $B = 10$ as well. Thus, the performance

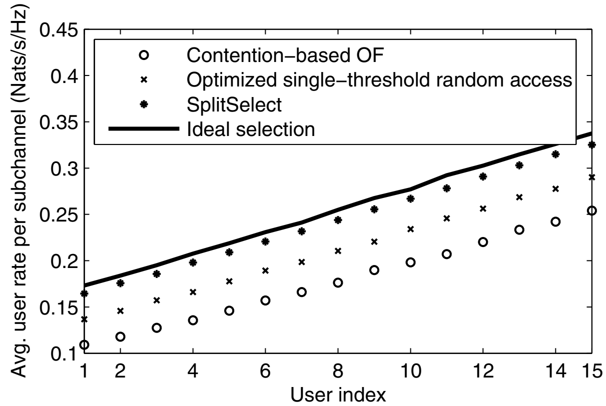


Fig. 8. Average user rates as a function of user index for the alternate proportional fair scheduler ($N = 15$, $S = 6$, and $B = 2$).

gains of SplitSelect arise primarily due to its speed and scalability and not because it is biased towards any particular scheduling policy.

B. User Prioritization

By rescaling the scheduling metrics differently, user priority can also be incorporated. For example, consider a system that consists of high priority (heavy traffic) users and low priority (low traffic) users. A high priority user is assigned a weight w_h while a low priority user is assigned a weight of w_l ($< w_h$). The scheduling metric is now defined as $S_{ij} = w_i H_{ij}$, where $w_i \in \{w_l, w_h\}$ is user i's weight and H_{ij} is the corresponding subchannel gain [33].[6]

Comparisons: We consider a system consisting of $N = 10$ users, of which users $6, \ldots, 10$ have high priority ($w_h = \frac{5}{30}$) and the remaining five have low priority ($w_l = \frac{1}{30}$). We assume i.i.d. subchannel gains. Fig. 9 plots the average rate per subchannel for different users for $B = 2$ slots per subchannel. SplitSelect, contention-based OF, and single-threshold based random access are compared with the reference case of ideal selection. We notice that the performance of SplitSelect is again closest to ideal selection. The benchmark schemes give lower-than-ideal throughputs to high priority users and vice versa. This is because of the more time required by them to assign the subchannels. Similar trends were also observed for $B = 10$.

VII. Conclusions

We developed a novel, fast, and distributed splitting-based multiple access algorithm called SplitSelect to opportunistically assign OFDMA downlink subchannels to their best users. Based on a scheduling criterion, each user contends

[5] In practice, when subchannel statistics are not available, $E[R_i]$ is evaluated using a moving window average. Traditionally, this averaging has been done by the BS since centralized feedback is assumed [3], [31], [32]. In SplitSelect, this can be done locally by the users themselves. Also, since the subchannels are identical, we note that $E[R_i] = SE[R_{ij}]$, where $E[R_{ij}]$ is the average rate of user i over subchannel j.

[6] The scheduling metrics of users are no longer i.i.d. even though their subchannel gains are. In order to run SplitSelect, the scheduling metrics need to be mapped to the interval $(0, N)$ while preserving their order across users and subchannels. This is achieved by computing the user metrics as $X_{ij} = N(1 - \bar{F}(S_{ij}))$, where $\bar{F}(x)$ is a composite CDF averaged across all the users' CDFs. It is given by $\bar{F}(x) = \frac{1}{\sum_{j=1}^{N} w_i} \sum_{i=1}^{N} w_i F_i(x)$, where F_i is the CDF of user i's scheduling metric. Note that this is a suboptimal formulation of what SplitSelect can handle since X_{ij} is, now, not uniform in $(0, N)$. Even so, its performance is very close to ideal. The same approach can also be used when users see statistically non-identical subchannel gains.

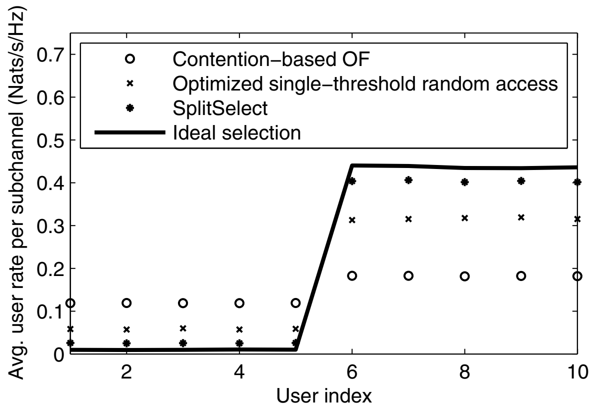


Fig. 9. Average user rates per subchannel as a function of user index for a scheduler that implements user prioritization. Users with index from 1 to 5 are low priority users ($w_l = 1/30$) and those with index from 6 to 10 are high priority users ($w_h = 5/30$) ($N = 10$, $S = 6$, and $B = 2$).

for its best unallocated subchannel. This framework was also extended to include fairness and user prioritization. A tractable asymptotic analysis led to simple update rules for the algorithm. Since the algorithm is based on a splitting approach, the time it requires to assign depends on the specific realizations of the subchannel gains. Even so, the average time required per subchannel is small irrespective of the number of users in the system. Further, it decreases as the number of subchannels increases. While SplitSelect is a splitting-based algorithm, it differs from conventional splitting algorithms in two interesting ways. First, the competing metrics, on the basis of which users contend, can get updated during the operation of the algorithm. This happens after a success, whence a subchannel gets allocated. Second, in SplitSelect, only a sub-interval of the transmission interval is half-split in the event of a collision.

SplitSelect *always* assigns each subchannel to its best user, which is unlike the single threshold based feedback schemes. It also has the desirable property that better subchannels get assigned earlier. Consequently, it outperforms several schemes proposed in the literature.

Appendix

A. Proof of Proposition 1

It is sufficient to show that the winner user, say i, does not lie in (L_k, R_k) in slot $k+1$ since the other users are always to the right of this user, *i.e.*, $m_j(k+1) \geq m_j(k) > R_k > m_i(k)$, $\forall j \neq i$.

Without loss of generality (w.l.o.g.), let the competing subchannel of winner user i in slot k be 1. Thus, $m_i(k) = X_{i1}$, which is uniformly distributed in $(0, N)$. Since k is a success slot

$$m_i(k+1) = \min_j \{X_{ij} : j \neq 1 \text{ and } j \text{ is not allocated}\}. \quad (12)$$

User i knows the set of allocated subchannels since the BS broadcasts the subchannel number assigned after every success. Hence, for a success slot k, we observe that $m_i(k)$ and $m_i(k+1)$ are independent. Further, $m_i(k+1)$ is the minimum of $\rho_k - 1$ i.i.d. RVs, each of which is uniformly distributed in $(0, N)$. Thus, the CDF of $m_i(k+1)$ is

$$F_{m_i(k+1)}(g) = 1 - \left(1 - \frac{g}{N}\right)^{\rho_k - 1}, \quad \forall g \in (0, N). \quad (13)$$

Let C_τ denote the event $\{m_i(k+1) \in (L_k, L_k + \tau)\}$ and D denote the event $\{m_i(k) \in (L_k, R_k), m_i(k+1) > m_i(k)\}$. Then, the probability of finding the new competing metric, $m_i(k+1)$, in $(L_k, L_k + \tau)$ given that slot k was a success with winner user i is

$$\mathbf{P}[C_\tau | D] = \frac{\mathbf{P}[C_\tau \cap D]}{\mathbf{P}[D]}.$$

We now evaluate $\mathbf{P}[D]$ and $\mathbf{P}[C_\tau \cap D]$ below.

1) $\mathbf{P}[D]$*:* Since $m_i(k)$ and $m_i(k+1)$ are independent when k is a success slot, we have

$$\begin{aligned} \mathbf{P}[D] &= \int_{L_k}^{R_k} \int_h^N \frac{1}{N} dF_{m_i(k+1)}(g)\, dh, \\ &= \frac{1}{\rho_k} \left[\left(1 - \frac{L_k}{N}\right)^{\rho_k} - \left(1 - \frac{R_k}{N}\right)^{\rho_k} \right]. \end{aligned} \quad (14)$$

2) $\mathbf{P}[C_\tau \cap D]$*:* If $\tau < R_k - L_k$, from the independence of $m_i(k)$ and $m_i(k+1)$, we have

$$\begin{aligned} \mathbf{P}[C_\tau \cap D] &= \int_{L_k}^{L_k+\tau} \int_h^{L_k+\tau} \frac{1}{N}\, dF_{m_i(k+1)}(g)\, dh, \\ &= \frac{1}{\rho_k} \left[\left(1 - \frac{L_k}{N}\right)^{\rho_k} - \left(1 - \frac{L_k + \tau}{N}\right)^{\rho_k} \right] \\ &\qquad - \frac{\tau}{N} \left(1 - \frac{L_k + \tau}{N}\right)^{\rho_k - 1}. \end{aligned} \quad (15)$$

And, for $\tau \geq R_k - L_k$, again from the independence of $m_i(k)$ and $m_i(k+1)$, we have

$$\begin{aligned} \mathbf{P}[C_\tau \cap D] &= \int_{L_k}^{R_k} \int_h^{R_k} \frac{1}{N}\, dF_{m_i(k+1)}(g)\, dh \\ &\quad + \int_{L_k}^{R_k} \int_{R_k}^{L_k+\tau} \frac{1}{N}\, dF_{m_i(k+1)}(g)\, dh, \\ &= \frac{1}{\rho_k} \left[\left(1 - \frac{L_k}{N}\right)^{\rho_k} - \left(1 - \frac{R_k}{N}\right)^{\rho_k} \right] \\ &\qquad - \frac{(R_k - L_k)}{N} \left(1 - \frac{L_k + \tau}{N}\right)^{\rho_k - 1}. \end{aligned} \quad (16)$$

Now, using (14), (15), and (16) to evaluate $\frac{\mathbf{P}[C_\tau \cap D]}{\mathbf{P}[D]}$, and taking the limit as $N \to \infty$, we get $\mathbf{P}[C_\tau | D] \overset{N \to \infty}{=} 0$. Hence, the result.

B. Proof of Proposition 2

If k is an idle slot in the non-collision mode, then all the users should lie to the right of R_k, *i.e.*, $m_i(k) > R_k, \forall i$. Further, since k is an idle slot, $m_i(k+1) = m_i(k)$, from (1). Hence, $m_i(k+1) > R_k, \forall i$. Thus, the average number of users, u, in $(R_k, R_k + \Delta_{k+1})$ in slot $k+1$ is

$$u = N\mathbf{P}[m_i(k+1) \in (R_k, R_k + \Delta_{k+1}) | m_j(k) > R_k, \forall j],$$

where i is an arbitrary user. Since the competing metrics of different users are i.i.d., we have

$$u = N\mathbf{P}[m_i(k+1) \in (R_k, R_k + \Delta_{k+1}) | m_i(k+1) > R_k], \\ = N\left[1 - \frac{\left(1 - \frac{R_k + \Delta_{k+1}}{N}\right)^{\rho_k}}{\left(1 - \frac{R_k}{N}\right)^{\rho_k}}\right]. \tag{17}$$

The last equality above follows because $m_i(k+1)$ is the minimum of the ρ_k unallocated metrics, each of which is uniformly distributed in $(0, N)$. Note here that $\rho_{k+1} = \rho_k$ since k is an idle slot. Taking limits as $N \to \infty$ on both sides of (17) yields the desired result.

C. Proof of Proposition 3

Let i be the winner user in slot k. For an arbitrary user $j \neq i$, let A'_j denote the event $\{m_j(k+1) \in (R_k, R_k + \Delta_{k+1})\}$ and B'_j denote the event $\{m_j(k) > R_k\}$. From Proposition 1, user i will not lie in $(V_k, R_k + \Delta_{k+1})$ in the next slot $k+1$ with probability 1 for large N. Therefore, for finite Δ_{k+1}, the average number of users in slot $k+1$ whose competing metrics lie in the interval $(R_k, R_k + \Delta_{k+1})$ is

$$(N-1)\mathbf{P}[A'_j | B'_j] = (N-1)\frac{\mathbf{P}[A'_i \cap B'_j]}{\mathbf{P}[B'_j]}. \tag{18}$$

Let subchannel 1 be, w.l.o.g., the competing subchannel of the winner user i in slot k, *i.e.*, $m_i(k) = X_{i1}$. Then,

$$m_j(k+1) = \min_l \{X_{jl} : l \neq 1 \text{ and } l \text{ is not allocated}\}.$$

Thus, X_{j1} and $m_j(k+1)$ are independent. The CDF of $m_j(k+1)$ is $F_{m_j(k+1)}(g) = 1 - \left(1 - \frac{g}{N}\right)^{\rho_k - 1}$, for $g \in (0, N)$, since it is the minimum of $\rho_k - 1$ RVs, which are uniformly distributed in $(0, N)$. Since $m_j(k+1)$ and X_{j1} are independent and X_{j1} is uniform in $(0, N)$, we have

$$\begin{aligned}\mathbf{P}[A'_j \cap B'_j] &= \int_{R_k}^{N} \int_{R_k}^{R_k + \Delta_{k+1}} \frac{1}{N} dF_{m_j(k+1)}(g)\, dh, \\ &= \left(1 - \frac{R_k}{N}\right)\left[\left(1 - \frac{R_k}{N}\right)^{\rho_k - 1} \right. \\ &\qquad \left. - \left(1 - \frac{R_k + \Delta_{k+1}}{N}\right)^{\rho_k - 1}\right].\end{aligned} \tag{19}$$

Further, $\mathbf{P}[B'_j] = \left(1 - \frac{R_k}{N}\right)^{\rho_k}$, since $m_j(k)$ is the minimum of ρ_k i.i.d. RVs, each of which is uniformly distributed in $(0, N)$. Thus, for $\Delta_{k+1} > 0$, $\mathbf{P}[A'_j | B'_j] = 1 - \frac{\left(1 - \frac{R_k + \Delta_{k+1}}{N}\right)^{\rho_k - 1}}{\left(1 - \frac{R_k}{N}\right)^{\rho_k - 1}}$. Substituting $\mathbf{P}[A'_j | B'_j]$ in (18) and taking the limit as $N \to \infty$, it can be shown that $(N-1)\mathbf{P}[A'_j | B'_j] \overset{N \to \infty}{=} (\rho_k - 1)\Delta_{k+1}$. Hence, the desired result follows.

D. Proof of Proposition 4

Let user 1 be, w.l.o.g., the winner user in slot k and let user 2 be the one that lies in the interval $(R_k, R_k + \Delta_k)$ during slot k. Then, the remaining $N-2$ users lie to the right of $R_k + \Delta_k$ during slot k. Let A_2 denote the event $\{m_2(k+1) \in (R_k, R_k + \Delta_{k+1})\}$ and B_2 denote the event $\{m_2(k) \in (R_k, R_k + \Delta_k)\}$. And for any user $j \geq 3$, let A_j be the event $\{m_j(k+1) \in (R_k, R_k + \Delta_{k+1})\}$ and B_j denote the event $\{m_j(k) > R_k + \Delta_k\}$. From Proposition 1, user 1 will not lie in $(V_k, R_k + \Delta_{k+1})$ in slot $k+1$ with probability 1 for large N. Thus, for finite $\Delta_{k+1} > 0$, the average number of users in slot $k+1$ that lie in the interval $(R_k, R_k + \Delta_{k+1})$ is

$$\begin{aligned}&\mathbf{P}[A_2 | B_2] + (N-2)\mathbf{P}[A_j | B_j] \\ &\qquad = \frac{\mathbf{P}[A_2 \cap B_2]}{\mathbf{P}[B_2]} + (N-2)\frac{\mathbf{P}[A_j \cap B_j]}{\mathbf{P}[B_j]},\end{aligned} \tag{20}$$

where the first term tracks the contribution from user 2 and the second from users $3, \ldots, N$.

Let 1 be the competing subchannel of the winner user 1 in slot k, *i.e.*, $m_1(k) = X_{11}$. Then, for all $i \geq 2$,

$$m_i(k+1) = \min_l \{X_{il} : l \neq 1 \text{ and } l \text{ is not allocated}\}.$$

Thus, X_{i1} and $m_i(k+1)$ are independent. The CDF of $m_i(k+1)$ is $F_{m_i(k+1)}(g) = 1 - (1 - \frac{g}{N})^{\rho_k - 1}$, for $g \in (0, N)$, since it is the minimum of $\rho_k - 1$ RVs that are uniformly distributed in $(0, N)$. And, X_{i1} is uniform in $(0, N)$. We now evaluate the probabilities $\mathbf{P}[A_2 | B_2]$ and $\mathbf{P}[A_j | B_j]$.

1) $\mathbf{P}[A_2 | B_2]$*:* If $\Delta_{k+1} < \Delta_k$, from the independence of $m_2(k+1)$ and X_{21}, we have

$$\begin{aligned}\mathbf{P}[A_2 \cap B_2] &= \int_{R_k}^{N} \int_{R_k}^{R_k + \Delta_{k+1}} \frac{1}{N}\, dF_{m_2(k+1)}(g)\, dh, \\ &= \left(1 - \frac{R_k}{N}\right)\left[\left(1 - \frac{R_k}{N}\right)^{\rho_k - 1} \right. \\ &\qquad \left. - \left(1 - \frac{R_k + \Delta_{k+1}}{N}\right)^{\rho_k - 1}\right].\end{aligned} \tag{21}$$

And for $\Delta_{k+1} \geq \Delta_k$, again from the independence of $m_2(k+1)$ and X_{21}, we have

$$\begin{aligned}\mathbf{P}[A_2 \cap B_2] &= \int_{R_k}^{N} \int_{R_k}^{R_k + \Delta_k} \frac{1}{N}\, dF_{m_2(k+1)}(g)\, dh \\ &\quad + \int_{R_k}^{R_k + \Delta_k} \int_{R_k + \Delta_k}^{R_k + \Delta_{k+1}} \frac{1}{N}\, dF_{m_2(k+1)}(g)\, dh.\end{aligned}$$

Simplifying, we get

$$\begin{aligned}&\mathbf{P}[A_2 \cap B_2] \\ &= \left(1 - \frac{R_k}{N}\right)\left[\left(1 - \frac{R_k}{N}\right)^{\rho_k - 1} - \left(1 - \frac{R_k + \Delta_k}{N}\right)^{\rho_k - 1}\right] \\ &+ \frac{\Delta_k}{N}\left[\left(1 - \frac{R_k + \Delta_k}{N}\right)^{\rho_k - 1} - \left(1 - \frac{R_k + \Delta_{k+1}}{N}\right)^{\rho_k - 1}\right].\end{aligned} \tag{22}$$

Likewise, we have

$$\begin{aligned}\mathbf{P}[B_2] &= \int_{R_k}^{N}\int_{R_k}^{R_k+\Delta_k}\frac{1}{N}\,dF_{m_2(k+1)}(g)\,dh \\ &\quad + \int_{R_k}^{R_k+\Delta_k}\int_{R_k+\Delta_k}^{N}\frac{1}{N}\,dF_{m_2(k+1)}(g)\,dh, \\ &= \left(1-\frac{R_k}{N}\right)\left[\left(1-\frac{R_k}{N}\right)^{\rho_k-1}-\left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k-1}\right] \\ &\quad + \frac{\Delta_k}{N}\left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k-1}. \end{aligned} \tag{23}$$

Using (21), (22), and (23) to evaluate $\mathbf{P}[A_2|B_2]$ and taking the limit as $N \to \infty$, we get

$$\mathbf{P}[A_2|B_2] \overset{N\to\infty}{=} \begin{cases} \frac{(\rho_k-1)\Delta_{k+1}}{\rho_k\Delta_k}, & \Delta_{k+1}\in[0,\Delta_k] \\ \left(1-\frac{1}{\rho_k}\right), & \Delta_{k+1}\geq\Delta_k \end{cases}. \tag{24}$$

2) $\mathbf{P}[A_j|B_j]$*:* Clearly, $\mathbf{P}[A_j|B_j] = 0$, for $\Delta_{k+1} < \Delta_k$, since we already know that $m_j(k) > R_k + \Delta_k$. From the independence of $m_j(k+1)$ and X_{j1}, for $\Delta_{k+1} \geq \Delta_k$, we have

$$\begin{aligned}\mathbf{P}[A_j\cap B_j] &= \int_{R_k+\Delta_k}^{N}\int_{R_k+\Delta_k}^{R_k+\Delta_{k+1}}\frac{1}{N}\,dF_{m_j(k+1)}(g)dh, \\ &= \left(1-\frac{R_k+\Delta_k}{N}\right)\left[\left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k-1}\right. \\ &\quad \left. -\left(1-\frac{R_k+\Delta_{k+1}}{N}\right)^{\rho_k-1}\right]. \end{aligned} \tag{25}$$

Since $m_j(k) = \min\{X_{j1}, m_j(k+1)\}$ is the minimum of ρ_k i.i.d. RVs, each of which is uniformly distributed in $(0, N)$, we have $\mathbf{P}[B_j] = \left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k}$. Using this and (25), we get $\mathbf{P}[A_j|B_j] = 1 - \frac{\left(1-\frac{R_k+\Delta_{k+1}}{N}\right)^{\rho_k-1}}{\left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k-1}}$, for $\Delta_{k+1} \geq \Delta_k$. Thus,

$$\begin{aligned}&(N-2)\,\mathbf{P}[A_j|B_j] \\ &\overset{N\to\infty}{=} \begin{cases} 0, & \Delta_{k+1}\in[0,\Delta_k] \\ (\rho_k-1)(\Delta_{k+1}-\Delta_k), & \Delta_{k+1}>\Delta_k \end{cases}. \end{aligned} \tag{26}$$

Taking the limit as $N \to \infty$ in (20) and using (24) and (26), the average number of users that transmit in the next slot is given by

$$\begin{aligned}&\mathbf{P}[A_2|B_2] + (N-2)\mathbf{P}[A_j|B_j] \\ &\overset{N\to\infty}{=} \begin{cases} \left(1-\frac{1}{\rho_k}\right)\frac{\Delta_{k+1}}{\Delta_k}, & \Delta_{k+1}\in[0,\Delta_k] \\ \left(1-\frac{1}{\rho_k}\right)+(\rho_k-1)(\Delta_{k+1}-\Delta_k), & \Delta_{k+1}>\Delta_k \end{cases}. \end{aligned}$$

3) Evaluation of p_1*:* p_1 is the probability of finding exactly one user lies in $(R_k, R_k + \Delta_k)$ in a success slot k that occurs in collision mode, given that at most two users lie in $(R_k, R_k + \Delta_k)$. The probability of a success slot with exactly one user lying in $(R_k, R_k+\Delta_k)$ is the probability that a winner user lies in (V_k, R_k), one user lies in $(R_k, R_k + \Delta_k)$, and the remaining $N-2$ users lie to the right of $R_k + \Delta_k$. In this case, w.l.o.g., let user 1 be the winner user, let user 2 lie in $(R_k, R_k + \Delta_k)$, and let the remaining users lie to the right of $R_k + \Delta_k$.

Similarly, the probability of a success slot k in collision mode, with exactly two users in $(R_k, R_k + \Delta_k)$, is the probability that the winner user, say 1, lies in (V_k, R_k), two users, say 2 and 3, lie in $(R_k, R_k + \Delta_k)$, and the remaining $N-3$ users lie to the right of $R_k + \Delta_k$.

Let W_1 denote the event $\{m_1(k) \in (V_k, R_k); m_2(k) \in (R_k, R_k + \Delta_k); m_j(k) > R_k + \Delta_k, \forall j \geq 3\}$ and let W_2 denote the event $\{m_1(k) \in (V_k, R_k); m_2(k), m_3(k) \in (R_k, R_k + \Delta_k); m_j(k) > R_k + \Delta_k, \forall j \geq 4\}$. Then, we have

$$p_1 = \frac{\mathbf{P}[W_1]}{\mathbf{P}[W_1]+\mathbf{P}[W_2]}. \tag{27}$$

From the fact that the competing metrics are the minimum of ρ_k i.i.d. RVs, each of which is uniformly distributed in $(0, N)$, and the fact that competing metrics across users are independent, we have, for $c = 1$ and 2,

$$\begin{aligned}\mathbf{P}[W_c] &= \binom{N-1}{c}\left(\left(1-\frac{R_k}{N}\right)^{\rho_k}-\left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k}\right)^{c} \\ &\quad\times\left(1-\frac{R_k+\Delta_k}{N}\right)^{\rho_k(N-1-c)}. \end{aligned} \tag{28}$$

Substituting $\mathbf{P}[W_1]$ and $\mathbf{P}[W_2]$ from (28) into (27) and taking limits as $N \to \infty$, we get $p_1 \overset{N\to\infty}{=} \frac{1}{1+\frac{\Delta_k\rho_k}{2}}$.

E. Brief Proof of Proposition 5

Since the analysis for this case draws heavily from Appendix D, we highlight its key steps in order to conserve space. Let user 1, w.l.o.g., be the winner user in slot k. And let users 2 and 3, w.l.o.g., be the ones that lie in the interval $(R_k, R_k + \Delta_k)$ in slot k. Then, the remaining $N-3$ users lie to the right of $R_k + \Delta_k$ during slot k. Let C_2 denote the event $\{m_2(k+1) \in (R_k, R_k + \Delta_{k+1})\}$ and D_2 denote the event $\{m_2(k) \in (R_k, R_k + \Delta_k)\}$. And, for a user $j \geq 4$, let C_j denote the event $\{m_j(k+1) \in (R_k, R_k + \Delta_{k+1})\}$ and let D_j denote the event $\{m_j(k) > R_k + \Delta_k\}$.

From Proposition 1, user i will not lie in $(V_k, R_k + \Delta_{k+1})$ in the next slot, $k+1$, with probability 1 for large N. Therefore, for a finite $\Delta_{k+1} > 0$, the average number of users in slot $k+1$ that lie in the interval $(R_k, R_k + \Delta_{k+1})$ is $2\mathbf{P}[C_2|D_2] + (N-3)\mathbf{P}[C_j|D_j]$, where the first term tracks the contributions from users 2 and 3 and the second from users $4, \ldots, N$. Proceeding along the same lines as in the previous case, for large N, $2\mathbf{P}[C_2|D_2]$ equals $2\left(1-\frac{1}{\rho_k}\right)\frac{\Delta_{k+1}}{\Delta_k}$, for $\Delta_{k+1} \in (0, \Delta_k)$, and $2\left(1-\frac{1}{\rho_k}\right)$, for $\Delta_{k+1} \geq \Delta_k$. And, $(N-3)\mathbf{P}[C_j|D_j] \overset{N\to\infty}{=} (\rho_k - 1)(\Delta_{k+1} - \Delta_k)$, for $\Delta_{k+1} \geq \Delta_k$, and is 0, otherwise. These expressions lead to (11). Further, $p_2 = 1 - p_1 \overset{N\to\infty}{=} \frac{\Delta_k\rho_k}{2+\Delta_k\rho_k}$.

REFERENCES

[1] S. Sesia, M. Baker, and I. Toufik, *LTE - The UMTS Long Term Evolution: From Theory to Practice*, 2nd edition. Wiley, 2011.

[2] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. 1995 IEEE ICC*, pp. 331–335.

[3] P. Viswanath, D. N. C. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, pp. 1277–1294, June 2002.

[4] D. Gesbert and M.-S. Alouini, "How much feedback is multi-user diversity really worth?" in *Proc. 2004 IEEE ICC*, pp. 234–238.

[5] P. Svedman, S. K. Wilson, L. J. Cimini, and B. Ottersten, "Opportunistic beamforming and scheduling for OFDMA systems," *IEEE Trans. Commun.*, vol. 55, pp. 941–952, May 2007.

[6] J. Leinonen, J. Hamalainen, and M. Juntti, "Performance analysis of downlink OFDMA resource allocation with limited feedback," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 2927–2937, June 2009.

[7] S. Sanayei and A. Nosratinia, "Opportunistic downlink transmission with limited feedback," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4363–4372, Nov. 2007.

[8] T. Tang and R. Heath, "Opportunistic feedback for downlink multiuser diversity," *IEEE Commun. Lett.*, vol. 9, pp. 948–950, Oct. 2005.

[9] T. Tang, C. Chae, R. Heath, S. Cho, and S. Yun, "Opportunistic scheduling in multiuser OFDM systems with clustered feedback," *Wireless Pers. Commun.*, vol. 52, pp. 209–225, Jan. 2010.

[10] R. Agarwal, V. Majjigi, Z. Han, R. Vannithamby, and J. Cioffi, "Low complexity resource allocation with opportunistic feedback over downlink OFDMA networks," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 1462–1472, Oct. 2008.

[11] J. So and J. M. Cioffi, "Feedback reduction scheme for downlink multiuser diversity," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 668–672, Feb. 2009.

[12] J. Chen, R. Berry, and M. Honig, "Limited feedback schemes for downlink OFDMA based on sub-channel groups," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 1451–1461, Oct. 2008.

[13] J.-G. Choi and S. Bahk, "Cell-throughput analysis of the proportional fair scheduler in the single-cell environment," *IEEE Trans. Veh. Technol.*, vol. 56, pp. 766–778, Mar. 2007.

[14] Y. Choi and S. Bahk, "Partial channel feedback schemes maximizing overall efficiency in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 1306–1314, Apr. 2008.

[15] M. Elkashlan, I. Collings, Z. Chen, and W. Krzymien, "Decentralized dynamic allocation of subchannels in multiple access networks," *IEEE Commun. Lett.,*, vol. 12, pp. 761–763, Oct. 2008.

[16] R. Agarwal, R. Vannithamby, and J. Cioffi, "Optimal allocation of feedback bits for downlink OFDMA systems," in *Proc. 2008 ISIT*, pp. 1686–1690.

[17] J. Chen, R. Berry, and M. Honig, "An adaptive limited feedback scheme for MIMO-OFDMA based on optimal stopping," in *Proc. 2008 Allerton Conf. Control, Commun., Comput.*, pp. 139–145.

[18] R. Agarwal, V. Abhishek, R. Vannithamby, and J. Cioffi, "Opportunistic feedback in OFDMA downlink," in *Proc. 2007 IEEE Globecom*, pp. 3220–3224.

[19] S. Bashar and Z. Ding, "Admission control and resource allocation in a heterogeneous OFDMA wireless network," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 4200–4210, Aug. 2009.

[20] V. Hassel, D. Gesbert, M.-S. Alouini, and G. Oien, "A threshold-based channel state feedback algorithm for modern cellular systems," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 2422–2426, July 2007.

[21] D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition. Prentice-Hall, Inc., 1992.

[22] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, pp. 505–515, Sep. 1979.

[23] B. Van Houdt and C. Blondia, "Analysis of an identifier splitting algorithm combined with polling (ISAP) for contention resolution in a wireless access network," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 2345–2355, Nov. 2000.

[24] X. Huang and C. Tellambura, "A MAC protocol with priority splitting algorithm for wireless ATM networks," in *Proc. 2000 IEEE VTC – Spring*, pp. 982–986.

[25] R. Garces and J. J. Garcia-Luna-Aceves, "Collision avoidance and resolution multiple access for multichannel wireless networks," in *Proc. 2000 IEEE INFOCOM*, pp. 595–602.

[26] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *Proc. 2004 IEEE INFOCOM*, pp. 1662–1672.

[27] V. Shah, N. B. Mehta, and R. Yim, "Splitting algorithms for fast relay selection: generalizations, analysis, and a unified view," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 1525–1535, Apr. 2010.

[28] S. Ko, S. Lee, H. Kwon, and B. G. Lee, "Mode selection-based channel feedback reduction schemes for opportunistic scheduling in OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 2842–2852, Sep. 2010.

[29] V. Shah, N. B. Mehta, and D. Bethanabhotla, "Performance of a fast, distributed multiple access based relay selection algorithm under imperfect statistical knowledge," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 3516–3527, Oct. 2011.

[30] S. M. Ross, *Introduction to Probability Models*, 9th edition. Academic Press, 2006.

[31] M. Ergen, S. Coleri, and P. Varaiya, "QoS aware adaptive resource allocation techniques for fair scheduling in OFDMA based broadband wireless access systems," *IEEE Trans. Broadcast.*, vol. 49, pp. 362–370, Dec. 2003.

[32] R. Almatarneh, M. Ahmed, and O. Dobre, "Performance analysis of proportional fair scheduling in OFDMA wireless systems," in *Proc. 2010 IEEE VTC – Fall*.

[33] Y. Ma, A. Leith, M.-S. Alouini, and X. Shen, "Weighted-SNR-based fair scheduling for uplink OFDMA," in *Proc. 2009 IEEE Globecom*, pp. 6276–6281.

[34] T. Tang, R. W. Heath, S. Cho, and S. Yun, "Opportunistic feedback for multiuser MIMO systems with linear receivers," *IEEE Trans. Commun.*, vol. 55, pp. 1020–1032, May 2007.

[35] A. Attar, N. Devroye, H. Li, and V. C. M. Leung, "A unified scheduling framework based on virtual timers for selfish-policy shared spectrum," in *Proc. 2010 IEEE ICC*.

A. Karthik received his Bachelor of Engineering degree in telecommunications engineering from the PES Institute of Technology, Bangalore, in 2006. He is currently pursuing his Ph.D. in the Electrical Communication Engineering Department at the Indian Institute of Science (IISc), Bangalore. During 2006–2007, he worked at CoreEL Technologies Pvt. Ltd., where he was involved in developing network interface cards on FPGAs. His research interests broadly include resource allocation in wireless networks and multiple access protocols.

Neelesh B. Mehta (S'98-M'01-SM'06) received his Bachelor of Technology degree in electronics and communications engineering from the Indian Institute of Technology (IIT), Madras in 1996, and his M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA in 1997 and 2001, respectively. He is now an Associate Professor in the Dept. of Electrical Communication Eng., Indian Institute of Science (IISc), Bangalore, India. Prior to joining IISc in 2007, he was a research scientist in the Wireless Systems Research Group in AT&T Laboratories, Middletown, NJ, USA from 2001 to 2002, Broadcom Corp., Matawan, NJ, USA from 2002 to 2003, and Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA from 2003 to 2007.

His research includes work on link adaptation, multiple access protocols, WCDMA downlinks, cellular systems, MIMO and antenna selection, energy harvesting networks, and cooperative communications. He was also actively involved in the Radio Access Network (RAN1) standardization activities in 3GPP from 2003 to 2007. He has served on several TPCs. He was a TPC co-chair for the WISARD 2010 and 2011 workshops, the National Conference on Communications (NCC) 2011, the Transmission Technologies track of VTC 2009 (Fall), and the Frontiers of Networking and Communications Symposium of Chinacom 2008. He was also the tutorials co-chair for SPCOM 2010. He has co-authored 29 IEEE journal papers, 55 conference papers, and three book chapters, and is a co-inventor in 16 issued US patents. He was an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2008 to 2011. He is now an Editor of IEEE WIRELESS COMMUNICATIONS LETTERS. He is serving as Director of Conference Publications in the Board of Governors of the IEEE Communications Society for the years 2012 and 2013. He is also an executive committee member of the IEEE Bangalore Section and the Bangalore Chapter of the IEEE Signal Processing Society.