Research Article

Collaborative Opportunistic Scheduling in Heterogeneous Networks: A Distributed Approach

Donglai Sun,¹ Yang Liu,² Jianhua Li,¹ and Yue Wu¹

¹ School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

² Department of EECS, University of Michigan, Ann Arbor, MI 48105, USA

Correspondence should be addressed to Jianhua Li; jhli888@sjtu.edu.cn

Received 14 March 2014; Revised 10 August 2014; Accepted 14 August 2014; Published 29 September 2014

Academic Editor: Baocang Ding

Copyright © 2014 Donglai Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider a collaborative opportunistic scheduling problem in a decentralized network with heterogeneous users. While most related researches focus on solutions for optimizing decentralized systems' total performance, we proceed in another direction. Two problems are specifically investigated. (1) With heterogeneous users having personal demands, is it possible to have it met by designing distributed opportunistic policies? (2) With a decentralized mechanism, how can we prevent selfish behaviors and enforce collaboration? In our research, we first introduce a multiuser network model along with a scheduling problem constrained by individual throughput requirement at each user's side. An iterative algorithm is then proposed to characterize a solution for the scheduling problem, based on which collaborative opportunistic scheduling scheme is enabled. Properties of the algorithm, including convergence, will be discussed. Furthermore in order to keep the users staying with the collaboration state, an additional punishment strategy is designed. Therefore selfish deviation can be detected and disciplined so that collaboration is enforced. We demonstrate our main findings with both analysis and simulations.

1. Introduction

Opportunistic scheduling of resource access within a multiuser network has been investigated during the last several years. This is very well motivated by examples and problems in wireless networks with shared medium and decentralized cognitive users, where how to effectively allocate the shared resource (e.g., frequency, time, energy, etc.) to users is essential for optimizing network's utilization. To answer this question, many research efforts have been made towards increasing system's total throughput (which in addition may increase service provider's revenue). Moreover, along with the development of centralized solutions for such classical scheduling problems [1, 2], studies of distributed mechanism have been emerging fast and are in great need for many practical scenarios. Corresponding results are provided in [3]. Within the category of distributed opportunistic scheduling, there are further two independent streams of researches. First of all with decentralized system, system designers aim at maximizing system's total throughput/benefits by designing particular scheduling policy for users; that is, the objective of

policy design is to track down the social optimal solutions. For example, in an ad hoc network users try to coordinate with each other to improve system's total throughput, while along the other direction networks with strategic users are considered. In such scenarios, each individual user only cares about their own benefits instead of the social welfare. Therefore the goal of research under this category is to seek the Nash equilibrium (N.E.) strategy under which no one would be willing to deviate unilaterally [4, 5].

An interesting observation from the above discussions is that individual requirement of each user has not been well studied in existing researches. Consider a system with multiple users: while each individual unit of the system tries to help achieve optimal performance (in fact this may not even be a goal), each of them may have an individual requirement over their own share of total network performance. This is however critical; otherwise users would have no incentive at all to help system reach the global optimum. This framework has a large area of applications. For example, consider a video streaming network. Different classes of subscribed users may demand different transmission qualities or bandwidth.

Although providing maximum overall throughput can make the largest profit for the network service provider, an unresolved problem is whether users would choose this provider's service or not, which is coupled with the satisfaction of their individual requirement. Our research is particularly motivated from resolving issues of scenarios where each user is associated with its own demand over service quality. In our work we consider a model with performance requirement captured by users' acquired throughput. Another interesting evaluation criteria and natural extension is considering each user's delay performance. This could be solved in the same way as presented in our paper; nevertheless it remains an interesting topic for future research.

According to the above discussions, we address the following issues in our research: (1) designing an opportunistic scheduling scheme so that channel diversities can be exploited as much as possible along with individual requirement for users being satisfied and (2) targeting a mechanism that can distinguish those unfeasible demands (i.e., requirements beyond the channel capacity). Moreover, considering the decentralized structure we considered, two more challenges have to be resolved: (1) the solution has to be a distributed one and (2) while selfish behaviors exist in a distributed network, coordination strategies have to be designed to prevent malicious competition and enforce collaboration. It is in this sense that we call it a distributed approach with collaborative opportunistic scheduling. More specifically in this paper, we first rigorously model and analyze multiuser network with individual throughput requirement. Then we try to tackle the distributed scheduling problem with goals satisfying each individual user's demand, which we will show later could be solved by stopping theory based threshold policies. We show by our designed algorithm optimal solutions can be effectively derived in a distributed way.

Since our scheduling policy requires collaborations and asks all users to follow a prespecified threshold strategy profile, problems arise for such a system with no coordination in the following sense: due to the selfish nature of strategic users, each of them could have incentive to deviate from the collaboration state (prespecified strategy), for example, when the assigned access strategy is not a Nash equilibrium (N.E.) which is in general the case. Therefore we proceed to the second step of our system design: try to enforce collaboration in a decentralized way. Since we are more interested in decentralized network, we will not assume any centralized coordinator which counters the essence of mechanism design for a distributed system. Instead in our work we show, by designing a punishment based mechanism enabled by group efforts, that users would be deterred from any selfish deviation. Theoretical analysis and simulation results are provided in our work to verify our design and claims.

The rest of the paper is organized as follows. Backgrounds and system model are provided in Section 2. Multiuser network with individual performance requirement is analyzed in Section 3, and a threshold enabled opportunistic scheduling policy is introduced to achieve the goal of optimizing users' throughput. A collaborative opportunistic scheduling policy and a corresponding iterative algorithm are presented in Section 4. Punishment policy and the strategy adopted for

enforcing collaboration are stated in Section 5. Simulation results are included in Section 6 with Section 7 concluding the paper.

2. Background and System Model

2.1. State of the Art. The idea of opportunistic scheduling originates from exploiting multiuser diversity which is firstly discussed by Tse in [6]. After preliminary studies showing system performance improvement can be obtained from diversity gaining process, for example, [7, 8], it has attracted the attentions of research community and then leads to the birth of opportunistic scheduling. Existing publications referring to this research domain provide efficient scheduling schemes based on different approaches. Scheduling for links with variable rates has been addressed in [1], which is also the first paper mentioning "opportunistic scheduling." A framework for applying this concept is proposed in this literature and authors also show how previous works of others can directly fit into this framework.

Recent studies concerning overall capacity optimization can be mainly categorized into two groups: (1) centralized mechanisms are used to solve scheduling problems in cellular communications systems and high-speed dedicated systems [9, 10]; (2) decentralized mechanisms, for example, DOS [4] and ADOS [11], are designed for wireless ad hoc networks. Diversified approaches are also designed to satisfy users' individual requirements with centralized mechanisms; for example, in [12–14] different methodologies are proposed to solve delay and throughput requirement for individual units within a system.

More specifically the research topics for opportunistic scheduling can be categorized into four branches based on different system model and performance evaluation criteria. Details can be found in Table 1. Due to the space limitation, only recent research results are listed. For global optimization problems, that is, optimization problems with respect to social utility, results have been well developed for both centralized and decentralized system. However research over investigating distributed solutions from individual user's perspective is relatively blank. Our works fill the lower right corner of Table 1; that is, we solve the scheduling problem concerning each user's individual performance requirement in a decentralized system.

2.2. System Model. In wireless networks, channel conditions vary from time to time. While users are not aware of channels' instantaneous condition, it is hard to decide whether and how they may accomplish each transmission, let alone to optimize channel utilization by designing and following specific channel access policies. Thus prepositive channel probing becomes a necessary step in opportunistic scheduling. Probing packets are used to detect current channel state, as well as for claiming medium access to avoid interferences (e.g., IEEE 802.11 CSMA/CA). To be more specific, users will send out a carrier sensing packet to reserve the transmission right over a certain channel before transmission. If the carrier sensing packet is correctly received, decoded, and

TABLE 1: A summary of former opportunistic scheduling researches.

Evaluation	Model	
	Centralized	Decentralized
Global optimization	Original OS [1]	DOS [4]
	HSDPA [9]	ADOS [11]
	CA-WS [10]	
Individual satisfaction	IS approach [12]	
	Dual approach [13]	<i>Our research</i>
	Virtual queue [14]	

acknowledged, the pair of users (transmitter/receiver pair, and we refer to it as link in the following part) successfully obtain the transmission right.

To simplify the above model and make our analysis tractable, we make the following assumptions. First of all, we consider networks consisting of multiple links (pairs of source/destination users, instead of users in an ad hoc network). Secondly, the system works in a discrete time fashion; that is, links try to access the channel and make decision at $\tau = 1, 2, \dots, T$. We assume, at each system time τ , that links will try to access channel with certain attempt rate. If one link successfully reserved the channel (or won the competition), it will reserve the channel for a certain period and transmit. Under this case, other links will back off during the reserved time and will *not* attempt to transmit. While multiple links try to access the channel at the same time slot, a collision is deemed to occur and none of the links succeeds in claiming the channel access. An example is shown in Figure 1.

The single-hop wireless network with N links is indexed by $\Omega = \{1, 2, \dots, N\}$. p_i denotes the probability of channel probing attempts of link i . Due to the dynamics of wireless environment, we model the time-evolving transmission rate of link i into a random variable X_i with p.d.f. $f_{X_i}(x)$, c.d.f. $F_{X_i}(x)$, and a finite upper bound $\bar{X}_i$. For the convenience of derivation, we denote by t the time duration of a slot and by t_p time duration of data transmission. For all individual requirements in the network, we assume that this information is publicly claimed in the whole network (which is important for payment in reality) and is denoted by $\mu = \{\mu_1, \mu_2, \dots, \mu_N\}$.

3. Achieving Optimal Individual Throughput with Threshold Enabled Opportunistic Scheduling

Following the system model, packets arrive at MAC layer of each link in a stochastic manner. In this case, random access can be a natural distributed solution. However, considering the individual throughput requirement, it is important to find out whether random access can fulfill these specific requirements or not. In this section, we first present a method for modeling each links' throughput so that analytical results can be derived to evaluate the performance of the widely adopted random access mechanism. Then by characterizing a threshold enabled policy, we show that, besides optimizing overall throughput of the system, links' individual throughput can also be optimized with threshold enabled policies.

FIGURE 1: An example of channel-aware scheduling in wireless network.

3.1. Links' Individual Throughput under Random Access. With channel probing probability p_i where $i = 1 \dots N$, the probability of successful channel access for link i can be given by $P_i = p_i \prod_{j \neq i} (1 - p_j)$. Then, the overall successful channel access probability P_s is given by

$$P_s = \sum_{i=1}^N P_i = \sum_{i=1}^N p_i \prod_{j \neq i} (1 - p_j). \quad (1)$$

Therefore for all the links, the average waiting time before a transmission is $t_r = t/P_s$. To calculate the throughput for each link, it is also important to calculate the average time for link i to wait before its own transmission, which is given by

$$t_r^i = \frac{t}{P_i} = \frac{t}{p_i \prod_{j \neq i} (1 - p_j)}. \quad (2)$$

At the same time while link i waits for the next transmission opportunity, some other links may succeed in channel contention and initiate transmission, in which case link i needs to back off and stay silent during the whole transmission (the same as the timer frozen mechanism in IEEE 802.11 DCF). To this end, the waiting time experienced by each single link is higher than t_r^i , and from a long-term aspect of view, it can be approximately calculated as

$$\tilde{t}_r^i = t_r^i + \frac{t_r^i - t_r}{t_r} \cdot t_p, \quad (3)$$

in which the second part is the expectation of potential transmission time for other links during the waiting period: again this is calculated from a mean-field's perspective. Thus, we can have the throughput for any link i described as follows:

$$S_{\text{ran}}^i = \frac{t_p}{t_p + \tilde{t}_r^i} \int_0^{\bar{X}_i} x \cdot f_{X_i}(x) dx. \quad (4)$$

Apparently when $\mu_i > S_{\text{ran}}^i$, link's individual requirement cannot be achieved with random access. Due to the fact that

random access is only an access resolution mechanism by nature, it can hardly achieve the network optimization. A following question is whether we can find a better scheduling scheme to solve the individually constrained problem.

3.2. Optimizing Individual Throughput with Distributed Threshold Policy. In this section, we design a distributed multithreshold opportunistic policy to achieve our goal: improving individual throughput as compared with random access. The distributed scheduling rule is a modified edition of random access and is explained as follows. After one link winning the channel competition, instead of an immediate transmission as in random access, a decision has to be made between the following two options.

- (1) When the link's current transmission rate is *higher* than its threshold, user will reserve the channel for a certain period and transmit. Under this case, other links will back off during the reserved time and will *not* attempt to transmit.
- (2) When the link's current transmission rate is *not higher* than its threshold, user will give up the transmission right and let the access competition for users within the system restart immediately.

Consider a threshold set denoted by $\mathbf{T} = \{T_1, T_2, \dots, T_N\}$. First, the probability of a successful transmission is given by

$$P_{t,s} = \sum_{i=1}^N P_i \cdot (1 - F_{X,i}(T_i)). \quad (5)$$

Therefore an average waiting time before a transmission is given by $t_w = t/P_{t,s}$. In a time period of $t_p + t_w$, a packet transmission takes place. For the average throughput of each link under threshold T_s , it can be derived following the same steps. As the probability of a successful transmission for link i is given by $P_i \cdot (1 - F_{X,i}(T_i))$, the actual average time for link i to wait can be derived as

$$t_w^i = \frac{t}{P_i \cdot (1 - F_{X,i}(T_i))}. \quad (6)$$

Again due to the back-off mechanism, the actual waiting time for link i becomes

$$\tilde{t}_w^i = t_w^i + \frac{t_w^i - t_w}{t_w} \cdot t_p, \quad (7)$$

in which the second part is the expected transmission time of other links during link i 's waiting period. Therefore the average throughput for link i is

$$S_{ave}^i(\mathbf{T}) = \frac{t_p}{t_p + \tilde{t}_w^i} \frac{\int_{T_i}^{\bar{X}_i} x \cdot f_{X,i}(x) dx}{1 - F_{X,i}(T_i)}. \quad (8)$$

Following the above derivations, we have Proposition 1.

Proposition 1. *For any link i , a threshold set $\mathbf{T} = \{T_1, T_2, \dots, T_N\}$ always exists for achieving optimal individual throughput S_{ave}^i .*

Proof. The proof can be found in Appendix A. $\square$

Remark. It is shown in Proposition 1 that threshold set $\mathbf{T}$ for achieving optimal individual throughput always exists. Therefore by using threshold based distributed scheduling policy, a larger range and a higher individual throughput requirement can be satisfied compared to random access scheduling.

4. Achieving Global Satisfaction: Collaborative Opportunistic Scheduling

With Proposition 1, we have proved that individual throughput can be optimized by threshold enabled opportunistic policy. In this section extra efforts are made towards establishing a mechanism to achieve individual satisfaction with opportunistic scheduling. Also as channel resource is limited, a further goal is to design a way of testing the feasibility of a requirement set μ .

4.1. Achieving Global Satisfaction. We first derive the thresholds to achieve global satisfaction. For link i , we can calculate the individual transmission rate under threshold set $\mathbf{T}$ from (8) as

$$S_{ave}^i(\mathbf{T}) = \frac{P_i \prod_{j \neq i} (1 - P_j)}{(t/t_p) + P_{t,s}} \int_{T_i}^{\bar{X}_i} x \cdot f_{X,i}(x) dx, \quad (9)$$

where $\mathbf{T}$ is the current threshold set. Therefore to have a set of thresholds for satisfying the individual requirement, it is obliged to have

$$S_{ave}^i(\mathbf{T}) \geq \mu_i, \quad \forall i \in \Omega. \quad (10)$$

As a result, the problem of deriving a feasible threshold can be treated as solving a system of nonlinear inequalities. If the system of inequalities is unsolvable under a requirement set, it is for sure that this set is not achievable. This can be used as a mechanism to distinguish between feasible and infeasible requirement sets. However, the direct calculation has a very high computation complexity and it is in general difficult to implement a heavy solver on embedded systems. We therefore focus on simpler solution in the sense of practical implementation.

The uniqueness of the solution is another interesting point worth addressing in this problem. Unfortunately, this is not necessarily true as a result of the complex distribution of different links' conditions. Even though it is possible to provide some sufficient conditions, with which unique solution can be established, it is out of interest of this paper since the major task is to achieve certain satisfaction level of individual throughput requirement.

4.2. An Iterative Algorithm. In the following part, we focus on an iterative algorithm for deriving one of the feasible threshold solution sets under individual requirement. Two rounds of iterations are adopted in the algorithm. The first round is at the whole threshold set level, of which at step k

we have a new threshold set $\mathbf{T}^{(k)}$. In the second round, we derive the threshold for link i with other links' thresholds, including those which have already been calculated in current step k , that is, $T_1^{(k)} \dots T_{i-1}^{(k)}$, and those that have not yet been renewed in current step, that is, $T_{i+1}^{(k-1)} \dots T_N^{(k-1)}$. While $k = 0$, we have the initial value for all the links as $T_1^{(0)} = \bar{X}_1, T_2^{(0)} = \bar{X}_2 \dots T_N^{(0)} = \bar{X}_N$. For convenience, we also denote $T_{-i} = \{T_1^{(k)}, T_2^{(k)}, \dots, T_{i-1}^{(k)}, T_{i+1}^{(k-1)}, \dots, T_N^{(k-1)}\}$. We state the algorithm as follows.

Algorithm Description

- (1) Initialize $k = 0$ and $T^{(0)}$ as above.
- (2) $k := k + 1$ and at step k , we define the following $g(T_i^*, T_{-i})$ as discussed in (A.5):

$$g(T_i^*, T_{-i}) = -T_i^* t - T_i^* P_{t,s} t_p + P_i t_p \int_{T_i^*}^{\bar{X}_i} x \cdot f_{X,i}(x) dx. \quad (11)$$

Then it is possible to have the optimized threshold T_i^* under the condition $g(T_i^*, T_{-i}) = 0$.

- (3) As presented in the proof of Proposition 1, the unique solution T_i^* of $g(T_i^*, T_{-i}) = 0$ exists and is the optimized threshold for having the best S_{ave}^i under T_{-i} . Then we can calculate the current S_{ave}^i with (8).
- (4) Based on the comparison between current S_{ave}^i and μ_i , we can have the following judgement:
 - (a) if $S_{ave}^i < \mu_i$, iteration terminated, no solution;
 - (b) if $S_{ave}^i = \mu_i$, update $T_i^{(k)} = T_i^*$;
 - (c) if $S_{ave}^i > \mu_i$, find the maximal solution for $S_{ave}^i(T_i^{(k)}, T_{-i}) = \mu_i$.
- (5) If the algorithm does not terminate, continue with the same procedure to calculate for the link $i + 1$ in step k . If it is already the last link in step k , move on to Step (2). If fixed points have arrived for all individual thresholds, the algorithm stops, and the current $\mathbf{T}^{(k)}$ is a solution to the scheduling problem.

Proposition 2. For a given and feasible set of requirements μ , the iterative algorithm returns a fixed point solution $\mathbf{T}^* = \{T_1^*, T_2^*, \dots, T_N^*\}$, so that $S_{ave}^i = \mu_i$, for all $i \in \Omega$.

Proof. The proof can be found in Appendix B. $\square$

Remark. (1) The convergence of the iterative algorithm is shown in Proposition 2. It is interesting that the convergence only appears when there exist solutions for the scheduling problem. Thus, if a threshold set can be derived from this algorithm, it is clear that the corresponding requirement set is achievable.

(2) The number of iterations to achieve the convergence is also important for the algorithm. Unfortunately, it is not always finite due to the nonmonotonicity of the output at each

step. However, the transmission rate is always digitalized in wireless network, and a certain level of accuracy is practicably enough. To show the performance of this part, experiments will be shown later.

(3) It can be observed that computing the threshold set requires the knowledge of all links in the network. Exchanging information with channel probing can be an easy solution for solving this problem; however it is also uncertain since fraud remains a potential problem. To this end, a more realistic solution is to apply an online learning algorithm. By observing previous channel contention and transmission, approximate channel state information can be practically learned. This mechanism is a well-investigated topic; an example can be found in Section IV.E of [4]; however this part is out of our scope in this paper and will not be discussed with details.

(4) Following the above discussions, the threshold set can be calculated in a distributed manner. By updating the local estimation of the overall network condition, approximate synchronization can be achieved while everyone in the network collaborates for the purpose of global satisfaction.

(5) By designing a multithreshold policy we have a collaborative opportunistic scheduling policy in the sense that everyone needs to follow a prespecified strategy instead of behaving in their own way.

4.3. Discussions on Algorithm Complexity. We provide insights and discussions on the algorithm's complexity as well as its convergence speed. We start with discussing its complexity. The major computation complexity comes from solving the fixed point equation in Step (1). Consider the following fixed point equation (via setting $g(T_i^*, T_{-i}) = 0$ and rearranging):

$$T_i^* = \frac{P_i t_p}{t + P_{t,s} t_p} \int_{T_i^*}^{\bar{X}_i} x f_{X,i}(x) dx. \quad (12)$$

Take the derivative of the right-hand side and we have its absolute value as follows:

$$\frac{P_i t_p}{t + P_{t,s} t_p} \cdot T_i^* f_{X,i}(T_i^*). \quad (13)$$

When $T_i^* f_{X,i}(T_i^*) \leq 1$ (e.g., with exponentially decayed tail), we have the convergence rate as $P_i t_p / (t + P_{t,s} t_p)$ for the fixed point equation. Therefore with a ϵ tolerance of solutions, we expect the algorithm convergence with $\mathcal{O}(\epsilon / (P_i t_p / (t + P_{t,s} t_p)))$ rounds.

We now turn to the convergence rate of the algorithm. The overall algorithm could be viewed as solving a system of fixed point equations. Therefore the convergence speed is determined by eigenvalues of the derivative matrix. The diagonal terms are determined as above. For the rest of the terms ((i, j)th entry of the derivative matrix with $i \neq j$) we have

$$\frac{P_i t_p}{(t + P_{t,s} t_p)^2} \cdot t_p \cdot P_i \cdot f_{X,i}(T_i). \quad (14)$$

When the number of links is large we know $P_{t,s} \gg P_i$; we could expect this term to be small. Therefore the derivative matrix could be viewed as a perturbed version of the diagonal matrix. Consider the following:

$$D + \eta \cdot \mathbf{1}^T \cdot \mathbf{1}, \quad (15)$$

where D is a diagonal matrix with diagonal entries being the following:

$$D_{i,i} = \lambda_i = \frac{P_i t_p}{t + P_{t,s} t_p} \quad (16)$$

while η is an upper bound on the (i, j) -entry with $i \neq j$. Therefore we know the maximum eigenvalue of $D + \eta \cdot \mathbf{1}^T \cdot \mathbf{1}$ is bounded as

$$\max_i \lambda_i + \eta \quad (17)$$

which is less than 1 if η is small enough. Thus with the accuracy region being ϵ (within $OPT \pm \epsilon$) we could expect the algorithm to converge to the tolerant region within $\mathcal{O}(\epsilon/(\max_i \lambda_i + \eta))$ rounds. And if systems are digitalized, when ϵ is small enough we converge to the optimal solution (due to the finite number of solutions).

4.4. Unachievable Requirement Set. It has been shown above that, with our mechanism, threshold policy can be used to achieve better individual throughput for links in wireless network. However, there is certainly a limitation in the sense that extreme cases can never be satisfied with limited channel resource. Therefore as mentioned before, another problem arises as how to distinguish achievable requirements from infeasible ones.

Interestingly, our iterative algorithm presented in the above section can be used directly for this problem.

Proposition 3. *For a given set of requirements μ , if the iterative algorithm terminates without converging, there is no solution set $\mathbf{T}^*$ for satisfying the requirement set μ .*

Proof. The proof is similar to the proof of Proposition 2. By forming contradictions, we can prove that termination is a special case where all the channel resources have been used up by even a smaller set of constraints. Due to space limitation, detailed proof is omitted for conciseness of presentation. $\square$

Remark. This proposition shows that the solution set is unachievable if the iterative algorithm terminates with no solution obtained. Therefore we have shown our iterative algorithm is able to fulfill both tasks: (1) deriving the thresholds and (2) distinguishing unfeasible regions.

4.5. Problem of Collaborative Opportunistic Scheduling. According to the above discussions, it can be concluded that individual thresholds can be calculated distributively by each link; and by performing the threshold enabled scheduling policy all links can achieve their throughput requirements. However, this satisfaction requires all links to be unselfish. If

one individual link deviates from the prescribed policy, no throughput guarantee can be made to anyone else. Moreover since we already characterized the average throughput per link mathematically with (8), we can easily check that, with other links' threshold being fixed, each link's throughput has a global maximizer with respect to its own threshold. Thus in a realistic scenario, links have incentives to deviate to this local maximizer which leads to unpredictable selfish behaviors. Then the network will end up with N.E. which is in general not the ideal condition of network (e.g., see results on price of anarchy), and individual requirement can hardly be guaranteed under such conditions. To this end, we design a punishment strategy to enforce distributed collaboration in the following section.

5. Enforcing the Collaboration

In this section we focus on designing decentralized mechanism that deter selfish behaviors (which we will detail later) of users. We assume links' objective is to maximize their discounted sum reward over infinite time horizon; that is,

$$\max \sum_{t=0}^{\infty} \delta^t \cdot S_{ave}^i(\mathbf{T}(t), \mathbf{p}(t)). \quad (18)$$

Here δ is the discount factor satisfying $0 < \delta < 1$ and we use δ to model a link's valuation over current and future reward; meanwhile we use this discounted sum to model system's long-term behavior. Moreover we explicitly write S_{ave}^i as a function of threshold set $\mathbf{T}$ and attempt probability $\mathbf{p}$. Denote our targeted threshold policy by $(\mathbf{T}^*, \mathbf{p})$. In the previous section we show a feasible threshold based scheduling policy can be effectively targeted to make the resource allocation fair among links in the sense that each one's individual throughput requirement is met. Furthermore in [4], N.E. has been shown to exist and efficient way of calculating N.E. has been proposed. We will reuse the results without restating for brevity.

Denote the N.E. of the access problem by $\hat{\mathbf{T}} = [\hat{T}_1, \dots, \hat{T}_N]$ and we design the following punishment based mechanism to avoid deviation. First we consider the two incentives under which links would deviate.

- (1) Deviate from attempt probability p_i to get more transmission air time.
- (2) Deviate from transmission threshold T_i^* to have either more airtime or better transmission quality, or both.

We start with describing our mechanism by defining the following states.

- (1) W_0 : start state: at W_0 links follow the strategy profile $(\mathbf{T}^*, \mathbf{p})$.
- (2) $W_i(t)$, $t = 1, 2, \dots, L$, $i \in \Omega$: punishment phases for link i . At state $W_i(t)$ links follow the specified strategy profile $(\hat{\mathbf{T}}(\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i}), (\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i}))$. Here $\hat{\mathbf{T}}$ is the N.E. described above while L is some finite positive integer which is the length of punishment phases. $\tilde{\mathbf{p}}_i$

is the punishment profile for all attempt rates at user i 's punishment phase.

Consider the following mechanism.

Mechanism Description of (M)

- (1) Links start at state W_0 . If all links follow the strategy profile $(\mathbf{T}^*, \mathbf{p})$ specified for W_0 , our targeted strategy profile will be kept playing at next time.
- (2) At any time t , if there happens to be some link, for example, i deviated from $(\mathbf{T}^*, \mathbf{p})$, starting from time $t + 1$, system goes into state $W_i(1)$ and play strategy profile $(\tilde{\mathbf{T}}(\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i}), (\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i}))$; that is, link i 's punishment phase is triggered. For detecting deviations, links essentially use past observed access statistics in a certain short period of time to estimate other links' current attempt rate. Though this would be imprecise in practice, for now we ignore this deficiency while focusing more on presenting the basic idea with a clean model.
- (3) At state $W_i(t)$, $t = 1, 2, \dots, L - 1$, if all links follow the punishment strategy, system goes to state $W_i(t + 1)$ at the next time point; at $W_i(L)$, again if all links follow the specified strategy, system goes back to state W_0 .
- (4) At state $W_i(t)$, $t = 1, 2, \dots, L$, if link i deviates from the specified strategy profile, system goes to state $W_i(1)$ at next time; that is, the punishment phase will repeat for link i . If link $j \neq i$ deviates, system goes to $W_j(1)$ and starts the punishment phase for link j . This restarting mechanism will deter users from deviating due to the fact that more punishment will be incurred.

Denote this mechanism by (M) and now we want to see whether there will be link deviating under (M); that is, we are interested in whether (M) is enforceable or not. Before proceeding to the proof we put some requirements on choosing over $(\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i})$, as well as intuitions. The basic idea is to make the punishment harsh enough to deter any deviation.

(1) $\tilde{\mathbf{p}}_{i,-i}$ should be large enough. Notice that with large enough $\tilde{\mathbf{p}}_{i,-i}$ (with each element close enough to 1) from (9) we have

$$S_i(\hat{\mathbf{T}}(\tilde{\mathbf{p}}_i), \tilde{\mathbf{p}}_i) \rightarrow 0. \quad (19)$$

This essentially follows the intuition that when some other link becomes excessively aggressive, link i will lose most of its airtime (throttled by other links) and thus results in a significant decrease of its achievable throughput. Therefore with appropriately chosen parameter set $\tilde{\mathbf{p}}$ we can get

$$S_i(\hat{\mathbf{T}}(\tilde{\mathbf{p}}_i), \tilde{\mathbf{p}}_i) < S_i(\mathbf{T}^*, \mathbf{p}). \quad (20)$$

(2) A second restriction over selecting $\tilde{\mathbf{p}}^*$ is that each link i would like to stay at others' punishment phase instead of its own; that is,

$$\begin{aligned} S_i(\hat{\mathbf{T}}(\tilde{p}_{j,j}, \tilde{\mathbf{p}}_{j,-j}), (\tilde{p}_{j,j}, \tilde{\mathbf{p}}_{j,-j})) \\ \geq S_i(\hat{\mathbf{T}}(\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i}), (\tilde{p}_{i,i}, \tilde{\mathbf{p}}_{i,-i})). \end{aligned} \quad (21)$$

Through simple algebras we can show the existence of such strategy profiles with respect to links' attempt rate; the details are thus omitted here.

(3) A third restriction on selecting $\tilde{\mathbf{p}}_i$ we put is as follows:

$$S_i(\hat{\mathbf{T}}(p_i = 1, \tilde{\mathbf{p}}_{i,-i}), (p_i = 1, \tilde{\mathbf{p}}_{i,-i})) < S_i^* \quad (22)$$

that is, $\tilde{\mathbf{p}}$ is large enough and the punishment is harsh.

Now we show, with appropriately chosen parameter, that (M) enforces the strategy profile $(\mathbf{T}^*, \mathbf{p})$ for all links.

Proposition 4. $(\mathbf{T}^*, \mathbf{p})$ is enforceable under mechanism (M).

Proof. The proof can be found in Appendix C. $\square$

6. Simulations

In this section, we provide simulations to validate our results. First of all, AWGN channel models are adopted to capture the stochastic nature of diversified links. Based on such a model, normalized SNR ρ is derived from the distance in a transmitter/receiver pair (detailed equation can be found in [15]). Thus, the transmission rate of one link can be given by the Shannon capacity equation as $R = \log(1 + \rho|H|^2)$ Nats/s/Hz, where H denotes the random channel gain with a complex Gaussian distribution. Then we know the c.d.f. of transmission rate for one link is as follows:

$$F(r) = 1 - \exp\left(-\frac{\exp(r) - 1}{\rho}\right). \quad (23)$$

Furthermore, it can be derived from (23) that the probability of having a transmission rate over 20 Nats/Hz/s is rather small. Without loss of generality, we assume all $\bar{X}$ to be 20 Nats/Hz/s. Also for the convenience of derivation, we assume that the packet transmission duration t_p is equal to $30t$. The following simulations are carried out in this scenario.

6.1. Numerical Results of Iterative Algorithm. Obviously, the number of iterations needed to achieve the fixed point is important in an iterative algorithm. We present an example in Table 2. The accuracy level is three decimal places. Two links with the normalized SNR $\rho_1 = 5$ and $\rho_2 = 40$ and channel probing probability $p_1 = p_2 = 0.5$ are used. The required throughput is set as $\mu = \{1, 1\}$. Thus, we have found that, after 5 rounds of iteration, the threshold set arrives at the fixed point. A further test is performed where individual requirement set μ is generated stochastically. With running 10,000 sample paths, statistical results are derived and shown in Table 3. The computation is rather quick, which further confirms that the computation complexity of our algorithm is light.

With the similar scenario, we present Figure 2 to show the effective area of our iterative algorithm. We change each link's individual throughput requirement in order to test the feasibility of different requirement set. According to the simulation results, the upper bound of random access is at the point $\{0.75, 1.5075\}$ (area A). No more throughput

TABLE 2: Convergence behavior of the iterative algorithm.

Link	s_0	s_1	s_2	s_3	s_4	s_5	s^*
Link 1	20	2.666	2.526	2.505	2.500	2.500	2.500
Link 2	20	4.795	4.743	4.734	4.733	4.733	4.733

s^* represent the final value of s after the convergence.

TABLE 3: Convergence speed of the iterative algorithm.

Rounds	2	3	4–6	7–9	10–15	Others
Percentage	11.1%	16.7%	25%	27.8%	16.7%	2.7%

FIGURE 2: The effective area.

requirement can be achieved by random access outside this area. Area C shows the requirement beyond the channel capacity. Area B is the area where our algorithm can provide its functionality.

6.2. Simulation Analysis of Iterative Algorithm. With the same scenario used in the numerical results, we simulate the scheduling scheme in our simulator developed in MATLAB according to [15]. From our results we show, with individual thresholds, that link 1 can achieve its required throughput most of the time, as shown in Figure 3(a). When the requirement is too large to be handled, that is, in the area C of Figure 2, threshold strategy fails to return a feasible solution, resulting in the missing part in the figure.

To show the advantage of our methodology, we provide Figures 3(b), 4(a), and 4(b) to show the comparison among different solutions with the existence of individual requirement. A brief result of link 2's throughput under different scheduling scheme is presented in the figures. The area without data is the requirement set that can not be achieved by the specific scheduling scheme. It is easy to see our framework is capable of handling more application scenarios compared to the other scheduling scheme.

Moreover, our framework is capable of achieving better throughput under more complicated network scenarios. In Figure 5, we show the result of a more complicated simulation. A network with 10 links is simulated in which we applied multiple scheduling schemes. We mainly focus on a specific link with bad channel condition, while, in random access and DOS, it is easy to find out that this link has to compete with 9 other links, and it suffers a very low throughput. However, if we consider the real individual requirement setting in this case, our methodology can achieve much better optimized throughput when the individual requirement of other links is low which is possible in a practical scenario. This is the case we want to solve in this paper.

More general simulation cases are also done for completeness of our study. In particular networks with multiple links are generated (normalized SNR is randomly selected), and our iterative algorithm is applied to achieve better network efficiency in these networks. Part of the results is shown in Figure 6. While individual throughput requirements are changing, the threshold is shifting correspondingly to fulfill the requirement. Link number is also another factor that affects the threshold selection. In Figure 6, we also show two cases with different link numbers.

6.3. Simulation Results for Punishment Strategy. To verify our collaboration and punishment strategy, we consider a two-link network. Without loss of generality, we assume that $\mu_1 = \mu_2 = 1$ and $\rho_1 = \rho_2 = 20$. Thus, by not deviating, it can be derived that both requirements can be achieved. Moreover, if links require more throughput, an extremum of 1.8092 Nats/s/Hz can be achieved simultaneously for both links according to our iterative algorithm. In the following, we take link 1 as a selfish example and set the attempt probability for punishing link 1 as $\tilde{p}_1^* = 0.9$, $\tilde{p}_2 = 0.95$, and for our finite horizon simulation we set simulation parameters as $\delta = 1$, $L = 10$, that is, discount fact as 1 and punishment phase length as 10.

Through simulation we observe that by deviating (link 1) both links' average throughputs become $S_{ave}^1 = 0.9284$, $S_{ave}^2 = 1.4898$. At link 1 and link 2 we see a performance decrease by 48.68% and 17.65%, respectively. By performance decrease, link 1 would be deterred from deviation. Moreover we observe that, under the punishment, link 1's baseline demand 1 is violated. It can therefore be concluded that our punishment strategy is effective for deterring selfish behavior.

7. Conclusion

In this paper, we have investigated the distributed scheduling problem under individual throughput requirement in order

FIGURE 3: Throughput performance comparison.

FIGURE 4: The throughput of link 2 in random access and DOS.

to balance individuals' resources within the networks. The idea of opportunistic scheduling has been applied so that it is possible to satisfy links' individual requirements as much as possible. Meanwhile since there is no centralized regulator, selfish behavior is designed to be punished according to our strategy; and thus collaboration can be enforced in a decentralized network. We show with analysis and simulations that our solutions can achieve the above goals.

Appendices

A. Proof of Proposition 1

Derived from (8), we can have

$$S_{\text{ave}}^i(\mathbf{T}) = \frac{t_p}{t_w^i + t_w^i \cdot P_{t,s} \cdot t_p} \int_{T_i}^{\bar{X}_i} x \cdot f_{X,i}(x) dx. \quad (\text{A.1})$$

To examine the influence of threshold set $\mathbf{T}$, we have to see how T_i and T_j ($j \neq i$) affect S_{ave}^i (which indicates there exist at least 2 links in the network). For all $T_j > 0$, we have

$$\frac{\partial S_{\text{ave}}^i}{\partial T_j} = \frac{P_i t_p^2 P_j f_{X,i}(T_j) \int_{T_i}^{\bar{X}_i} x \cdot f_{X,i}(x) dx}{(t + t_p P_{t,s})^2} > 0, \quad \text{a.s.} \quad (\text{A.2})$$

It is clear that the existence of T_j can provide better effective transmission rate for link i . Therefore the only remaining problem is on T_i . As we have

$$\frac{\partial P_{t,s}}{\partial T_i} = -P_i f_{X,i}(T_i), \quad (\text{A.3})$$

FIGURE 5: Throughput comparison among different approaches.

FIGURE 6: Threshold calculation in multiple scenarios.

then

$$\frac{\partial S_{\text{ave}}^i}{\partial T_i} = \frac{P_i t_p f_{X,i}(T_i) \left[-T_i t - T_i P_{t,s} t_p + P_i t_p \int_{T_i}^{\bar{X}_i} x \cdot f_{X,i}(x) dx \right]}{(t + t_p P_{t,s})^2}. \quad (\text{A.4})$$

In this case, we examine $g(T_i)$ as follows:

$$g(T_i) = -T_i t - T_i P_{t,s} t_p + P_i t_p \int_{T_i}^{\bar{X}_i} x \cdot f_{X,i}(x) dx. \quad (\text{A.5})$$

It can be shown that

$$g'(T_i) = -t - P_{t,s} t_p < 0, \quad (\text{A.6})$$

which implies that $g(T_i)$ is strictly decreasing. Obviously, the maximizer is at $g(T_i) = 0$; and if $g(0) > 0$, there is always a maximizer T_i . As we have

$$g(0) = P_i t_p \int_0^{\bar{X}_i} x \cdot f_{X,i}(x) dx > 0, \quad (\text{A.7})$$

the existence of maximizer T_i has been proved, and moreover $T_i > 0$.

B. Proof of Proposition 2

First, we prove that the iterative algorithm converges to a certain point. As we have shown in Appendix A, $g(T_i)$ is strictly descending. Following the iterative algorithm, we can prove that $T_i^{(k+1)} \leq T_i^{(k)}$. Also, it is clear that all threshold sets $\{T\}$ have lower bounds according to the individual requirement μ and the optimized algorithm. Thus, if the algorithm does not terminate, it converges to a certain threshold set.

Second, we prove that the iteration can achieve a fixed point according to the algorithm, and this fixed point must satisfy the following two terms for all i :

$$g(T_i) \leq 0; \quad (\text{B.1a})$$

$$S_{\text{ave}}^i(T) = \mu_i. \quad (\text{B.1b})$$

Next we prove that, for a threshold set satisfying (B.1a) and (B.1b), it is a fixed point of the algorithm. We assume that $T^* = \{T_1^*, T_2^*, \dots, T_N^*\}$ is a threshold set for the problem. As we can get the unique $\tilde{T}_1^*$ from $g(\tilde{T}_1^*, T_{-1}) = 0$, obviously,

$$S_{\text{ave}}^i(\tilde{T}_1^*, T_{-1}) \geq S_{\text{ave}}^i(T_1^*, T_{-1}) = \mu_1. \quad (\text{B.2})$$

If we have $S_{\text{ave}}^i(\tilde{T}_1^*, T_{-1}) = \mu_1$, then $\tilde{T}_1^* = T_1^*$. If $S_{\text{ave}}^i(\tilde{T}_1^*, T_{-1}) > \mu_1$, according to (B.1a), we can find the unique $T_1 > \tilde{T}_1^*$, so that

$$S_{\text{ave}}^i(T_1, T_{-1}) = S_{\text{ave}}^i(T_1^*, T_{-1}). \quad (\text{B.3})$$

Thus, we have $T_1 = T_1^*$, and we do not change T_1^* . Similarly, we do not change $T_2^*, \dots, T_N^*$ under the algorithm. It is a fixed point for this iteration. We have proved the existence of the fixed point.

Then we prove that if T^* is a fixed point, it has to satisfy (B.1a) and (B.1b). We assume that issue (B.1b) cannot be satisfied; then we have two possible situations:

- (1) $\exists i$, s.t. $S_{\text{ave}}^i < \mu_i$; then the iteration should terminate;
- (2) $\exists i$, s.t. $S_{\text{ave}}^i > \mu_i$; then we can find the unique $T_i > T_i^*$ that can satisfy $S_{\text{ave}}^i = \mu_i$.

Obviously, it contradicts the assumption. Thus, item (B.1b) should always be satisfied.

Then assuming that issue (B.1a) can not be satisfied, we have the following situation: $\exists i$, s.t. $T_i^* < S_{\text{ave}}^i$. We can always find a unique T_i , so that we can have $T_i > T_i^*$ and $T_i^* = S_{\text{ave}}^i$. Again, it contradicts the assumption. Thus, item (B.1a) should always be satisfied.

Furthermore, we show that the convergence point of the algorithm can satisfy (B.1a) and (B.1b).

- (1) For item (B.1a), it always holds as required in the algorithm.
- (2) For item (B.1b), as we always have $S_{\text{ave}}^i = \mu_i$, it is always satisfied. Otherwise, the iteration terminates.

Thus, this algorithm converges to a fixed point according to the iterative algorithm. It is obvious that this fixed point is a solution as it satisfies (B.1a) and (B.1b). Thus, we have proved the proposition.

C. Proof of Proposition 4

To prove the enforceability of $(\mathbf{T}^*, \mathbf{p})$ we need to check no link would like to go for a one-step deviation at any state any time. First consider links at state W_0 . Without losing generality consider link i . Denote the utility from following the prespecified strategy profile for link i by S_i^* . Then by following the specified strategy at W_0 , the long run accumulated utility for link i is given by

$$S_i(W_0) = \delta^0 \cdot S_i^* + \dots + \delta^t \cdot S_i^* + \dots = \frac{1}{1-\delta} S_i^*. \quad (\text{C.1})$$

Next check the utility for link i by deviating to another profile (T_i', p_i') . Remember that, due to the finiteness of X_i and p ($0 \leq p \leq 1$), we have some finite positive number M_i such that

$$S_i((T_i', p_i'), (\mathbf{T}, \mathbf{p})_{-i}) \leq M_i. \quad (\text{C.2})$$

Also denote the utility of link i of playing $(\hat{\mathbf{T}}, (\tilde{p}_i, \tilde{\mathbf{p}}_{-i}))$ at punishment phase $W_i(t)$ by S_i^p . Then by a one-step deviation we know the aggregated utility for link i is

$$S_i^d(W_0) \leq M_i + \delta \cdot S_i^p + \delta^2 \cdot S_i^p + \dots + \delta^L \cdot S_i^p + \delta^{L+1} \cdot S_i^* + \dots. \quad (\text{C.3})$$

Then we have

$$S_i(W_0) - S_i^d(W_0) \geq (S_i^* - M_i) + (S_i^* - S_i^p) \cdot \frac{\delta - \delta^{L+1}}{1 - \delta}. \quad (\text{C.4})$$

Since we have $S_i^* - M_i \leq 0$, $S_i^* - S_i^p > 0$, with a large enough δ and L we could have

$$\frac{\delta - \delta^{L+1}}{1 - \delta} \rightarrow L, \quad (S_i^* - S_i^p) \cdot L \geq M_i - S_i^*. \quad (\text{C.5})$$

Thus we proved no one would deviate at state W_0 .

Next we analyze the punishment phases. Consider $W_i(t)$. First notice that obviously all links at any punishment phase with any stage $W_j(t)$, for all $j \in \Omega$, $t = 1, 2, \dots, L$, have no incentive to deviate with the transmission threshold since the threshold is a N.E. and no one could gain by deviation. Therefore we only need to consider links deviating with attempt rate $\mathbf{p}$.

First consider link i . Denote by $S_i(W_i)$ the utility link i can get by following W_i . By increasing its attempt rate, link i will increase its utility. But again due to the finiteness of p_i we have an upper bound for this one-step increase and we denote it by M_i' . Then suppose, at stage t , that link i 's utility by following specified strategy is given by

$$S_i(W_i(t)) = S_i(W_i(t)) + \delta \cdot S_i(W_i(t)) + \dots + \delta^{L-t-1} S_i(W_i(t)) + \delta^{L-t} S_i(W_0) + \dots, \quad (\text{C.6})$$

and by deviating we have

$$S_i^d(W_i(t)) \leq M_i' + \delta \cdot S_i(W_i(t)) + \dots + \delta^L S_i(W_i(t)) + \delta^{L+1} S_i(W_0) + \dots. \quad (\text{C.7})$$

Then take the difference of the above two terms; we have

$$S_i(W_i(t)) - S_i^d(W_i(t)) \geq S_i(W_i(t)) - M_i' + \delta \cdot (S_i(W_0) - S_i(W_i(t))). \quad (\text{C.8})$$

As $0 < M_i' - S_i(W_i(t)) < S_i(W_0) - S_i(W_i(t))$ with a large enough δ we have $S_i(W_i(t)) - M_i' + \delta \cdot (S_i(W_0) - S_i(W_i(t))) > 0$.

The last step is to check whether link $j \neq i$ would deviate at state $W_i(t)$ or not. As we already put constraints on $\{\tilde{\mathbf{p}}_i\}_i$ such that each link would rather stay at another link's punishment phase instead of their own, hereby link j would not deviate. Enforceability is therefore established.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the International Researcher Exchange Project of the National Science Foundation of China and the French Centre National de la Recherche Scientifique (NSFC-CNRS) under the Grant no. 61211130104 and the National Science Foundation of China under Grants nos. 60932003 and 61271220.

References

- [1] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, no. 4, pp. 451-474, 2003.
- [2] A. Gyasi-Agyei, "Multiuser diversity based opportunistic scheduling for wireless data networks," *IEEE Communications Letters*, vol. 9, no. 7, pp. 670-672, 2005.

- [3] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 1671–1688, 2013.
- [4] D. Zheng, W. Ge, and J. Zhang, "Distributed opportunistic scheduling for ad hoc networks with random access: an optimal stopping approach," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 205–222, 2009.
- [5] S. Tan, D. Zheng, J. Zhang, and J. Zeidler, "Distributed opportunistic scheduling for ad-hoc communications under delay constraints," in *Proceedings of the IEEE (INFOCOM '10)*, pp. 1–9, San Diego, Calif, USA, March 2010.
- [6] D. Tse, "Multiuser diversity in wireless networks," 2001.
- [7] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 321–331, April 2003.
- [8] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM '04)*, pp. 121–132, New York, NY, USA, September 2004.
- [9] H. Al-Zubaidy, I. Lambadaris, and J. Talim, "Optimal scheduling in high-speed downlink packet access networks," *ACM Transactions on Modeling and Computer Simulation*, vol. 21, no. 1, article 3, 2010.
- [10] S. Liu, L. Ying, and R. Srikant, "Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics," *IEEE/ACM Transactions on Networking*, vol. 19, no. 4, pp. 1057–1070, 2011.
- [11] A. Garcia-Saavedra, A. Banchs, P. Serrano, and J. Widmer, "Distributed opportunistic scheduling: a control theoretic approach," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 540–548, March 2012.
- [12] H. Kim and G. de Veciana, "Losing opportunism: evaluating service integration in an opportunistic wireless system," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 982–990, Anchorage, Alaska, USA, May 2007.
- [13] J.-A. Kwon, B.-G. Kim, and J.-W. Lee, "A unified framework for opportunistic fair scheduling in wireless networks: a dual approach," *Wireless Networks*, vol. 16, no. 7, pp. 1975–1986, 2010.
- [14] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proceedings of the IEEE INFOCOM*, pp. 1728–1736, Shanghai, China, April 2011.
- [15] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*, vol. 1, 2002, <http://books.google.com/books?id=amhNM01OKCUC>.