Extended Proportional Fair Scheduling for Statistical QoS Guarantee in Wireless Networks

Neung-Hyung Lee, Jin-Ghoo Choi, and Saewoong Bahk

Abstract: **Opportunistic scheduling provides the capability of resource management in wireless networks by taking advantage of multiuser diversity and by allowing delay variation in delivering data packets. It generally aims to maximize system throughput or guarantee fairness and quality of service (QoS) requirements. In this paper, we develop an extended proportional fair (PF) scheduling policy that can statistically guarantee three kinds of QoS. The scheduling policy is derived by solving the optimization problems in an ideal system according to QoS constraints. We prove that the practical version of the scheduling policy is optimal in opportunistic scheduling systems. As each scheduling policy has some parameters, we also consider practical parameter adaptation algorithms that require low implementation complexity and show their convergences mathematically. Through simulations, we confirm that our proposed schedulers show good fairness performance in addition to guaranteeing each user's QoS requirements.**

Index Terms: **Convergence, convex optimization, opportunistic scheduler, proportional fairness, quality of service (QoS) constraint, utility.**

I. INTRODUCTION

Recently, in wireless systems, broadband and high frequency have been implemented to meet the high bandwidth demands of each user. Wireless system targets accommodating various user applications. To make this happen, we use packet scheduling as a means of resource management which is a hot issue these days. In contrast to wire-lined systems of fixed channel rates, wireless systems have channels of time-varying features, which enables opportunistic scheduling. Third generation (3G) cellular systems such as CDMA2000 1xEV-DO evolved from high data rate (HDR), and the high speed downlink packet access (HSDPA) of universal mobile telecommunications systems (UMTSs) have deployed their own schedulers that run on the top of some mechanism that measures and gathers the channel state of each user.

Manuscript received December 10, 2008; approved for publication by Felipe A. Cruz-Perez, Division II Editor, March 25, 2010.

Part of this paper was presented in "Opportunistic Scheduling for Utility Maximization under QoS Constraints," *IEEE PIMRC*, Berlin, Germany, 2005.

This work was partly supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 09-043 (0423-20090043)) and the Ubiquitous Computing and Network (UCN) Project, the Ministry of Knowledge and Economy (MKE) Knowledge and Economy Frontier R&D Program in Korea as a result of UCNs subproject 10C2-C1-20S.

N.-H. Lee is with the DMC R&D Center, Samsung Electronics, 416, Maetan-3dong, Yeongtong-gu, Suwon-City, Gyeonggi-do, 443-742, Korea, email: nhlee.lee@samsung.com.

J.-G. Choi is the corresponding author with the Department of Information and Communication Engineering, Yeungnam University, email: jchoi@yu.ac.kr.

S. Bahk is with the School of Electrical Engineering and INMC, Seoul National University, Seoul, Korea, email: sbahk@netlab.snu.ac.kr.

The primary purpose of scheduling is to improve wireless channel efficiency by first serving users that have good channel conditions. Even though such a scheme shows good performance in terms of throughput, it has a fairness problem. Some users located near the cell boundary have bad channel conditions compared to others close to the base station. If the objective is throughput maximization, bad channel users have a lower chance of selection than good channel users. Therefore, fairness is an important performance measure that also must be considered.

A fair scheduler attempts to give each user an equal chance, insofar as it is possible, regardless of channel condition. The proportional fair (PF) scheduler in [1]–[3] achieves proportional fairness while taking advantage of multiuser diversity [4]. Due to its simplicity, it was adopted in the CDMA2000 1xEV-DO system. In some sense, fairness is a factor of quality of service (QoS) because it guarantees some portion of resources to each user. The fair scheduling in [5] has the objective of meeting each user's throughput target. In a strict sense, it needs to be applied for resources remaining after guaranteeing each user's QoS.

Delay is another QoS measure. The objectives of scheduling in [6] and [7] are to guarantee the delay bound of each user. In [8], the satisfaction of each user's fixed deadline and the maximization of achievable revenue have been considered together. These schedulers have the explicit delay constraint for opportunistic scheduling. If a scheduler continuously provides a fixed level of throughput for each user, it can guarantee each user's delay bound. In [9], the throughput requirement was expressed as a form of effective capacity.

In addition to throughput fairness and delay, temporal fairness, minimum throughput, and utilitarian fairness are considered as QoS metrics. After the QoS metrics are introduced in [10], they are commonly accepted QoS requirements in wireless scheduling. In [10], an opportunistic scheduling algorithm for guaranteeing minimum throughput has been proposed. It has the objective of utility maximization, and considers other QoS requirements such as temporal fairness and utilitarian fairness. Temporal fairness aims to allocate a fixed portion of time to each user while the utilitarian fairness aims to provide some portion of utility for each user. The temporal share fairness leads each user to get his/her minimum time share. For example, if there are 50 scheduling opportunities and user 1 wants 20% of the temporal share, the scheduler chooses user 1 more than 10 times. This metric targets achieving equality of opportunity like a weighted round robin. The minimum throughput requirement leads each user to receive some minimum throughput regardless of its location. In this case, a bad channel user needs to be scheduled more often than others in a good channel if his/her minimum throughput requirements are the same. Utilitarian fairness or

throughput-share requirement demands the scheduler to allocate each user more than a minimum portion of the total throughput. A high priority user who requires high throughput-share is scheduled more often compared to a low priority user. These three QoS requirements are independent, and sometimes each user may require different combinations of these. However, the scheduling policies in [10] can not support multiple QoS types for each user.

The utility is a useful tool that can be used for fairness management. The utility represents the satisfaction level of each user, so we can define the utility in many ways. In [1], it was shown that the PF scheduler maximizes the utility that has a logarithmic form of average throughput. In [11] and [12], the utilities are defined as a function of instantaneous signal to interference ratio (SIR) and signal to interference and noise ratio (SINR), respectively. In [13] and [14], a function of average throughput is used as utility while the instantaneous rate is used for mathematical analysis. In [15], the utility is defined as a concave function of average throughput and the convergence of PF scheduling is discussed. In [16], the utility is given as a function of instantaneous channel rate multiplied by the average serviced resource. In [17], a utility is given as a function of bandwidth and its performance is evaluated through simulations. In [18] and [19], a weighted PF scheduling algorithm and gradient based scheduling algorithm was proposed from utility maximization formulation. In [20], gradient algorithm with minimum/maximum rate constraints was proposed for scheduling, and in [21], the optimality of the gradient scheduling algorithm was proved asymptotically.

As utility, we can consider two types of functions. One is the function of average throughput. If we apply this to a downlink scheduling system, we can compute the utility after a sufficient amount of time. The PF scheduling corresponds to this type and uses a logarithmic function. The other is the function of throughput for one timeslot. This is referred to as slot utility. The scheduling policy in [10] uses this type. We can easily verify that these two types result in different utility values when applied for a finite number of time slots. For example, assume that the throughputs of a user for four consecutive slots are 4, 2, 9, and 1 Mbps, respectively. If the utility has a form of the square root of throughput, the utility of the former type is $\sqrt{(4+2+9+1)/4} = 2$ and that of the latter type is $(\sqrt{4}+\sqrt{2}+\sqrt{9}+\sqrt{1})/4 \simeq 1.85$. In our opportunistic scheduling, we use the former type.

In this paper, we deal with the derivation and the optimality proof of extended PF scheduling that guarantees three types of QoS metric statistically, and the convergence of proposed parameter updating algorithm. Although the QoS metrics of our proposed scheduling is the same as those in [10], there is a difference between them. That is, our proposed scheduling has the generalized form of PF scheduling, which is the most popular type for opportunistic scheduling, while Liu's proposal is not applicable for the PF type scheduler. Our proposal can also support the combined type of QoS metrics. In this paper, we first show the relationship between our scheduling and PF scheduling, and prove the optimality of our scheduling policies through mathematical analysis. Then, we develop an algorithm for updating parameter values for our scheduling policies based on the stochastic approximation in [23] and prove its convergence. These parameters act like weights that help our schedulers to meet each user's QoS requirements by compensating for time-varying channel conditions.

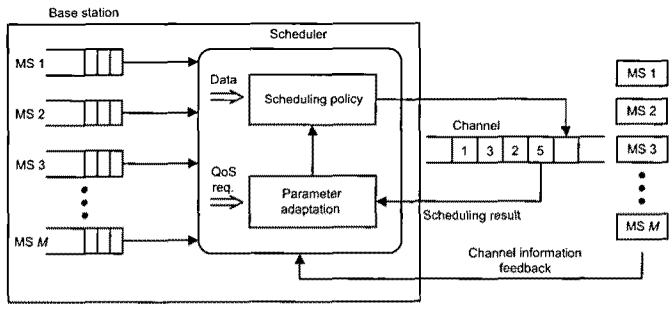

Fig. 1. Framework of wireless opportunistic scheduling.

Our proposed scheduling policies have several merits. First, they can support different utility functions for different users. Although it is a natural assumption that each user can have his/her own utility function, some existing schedulers assume that each user has the same utility function for the convenience of analysis or implementation. Second, our policies can also support combined types of QoS requirements for each user. Lastly, our policies are simple enough for easy implementation. Compared to the existing PF scheduler, our schedulers require an additional step of parameter adaptation, whose overhead is light.

The rest of our paper is organized as follows. In Section II, we derive an extended PF scheduling policy and prove the optimality of the policy. An adaptive algorithm to update scheduling parameters for optimal schedulers is also presented and their convergence is dealt with. Simulation results are given in Section III, followed by the conclusion in Section IV.

II. OPTIMAL OPPORTUNISTIC SCHEDULER

In this section, we consider optimal opportunistic scheduling in a single channel downlink system that uses time division multiplexing. An example system is given in Fig. 1. In our model, there are M mobile stations (MSs) and the base station (BS) sends the pilot signal periodically. We assume that proper admission control is provided, though it is not shown in Fig. 1. Each MS measures its channel gain and feeds it back to the BS. Each MS has its own utility function and reports the form of utility function to the scheduler before the scheduling service starts. The scheduler selects a user who will be served at the next slot. Scheduling parameter values are updated according to each user's channel condition, scheduling results, and QoS requirements. To design an efficient and stable scheduling system, the optimality of scheduling policy part and the convergence of parameter adaptation part should be guaranteed.

We consider two kinds of scheduling schemes. One is an off-line optimal scheduling and the other is opportunistic scheduling. In an off-line scheduling scheme, unrealistic assumptions of knowing the channel rates beforehand are used because its purpose is the mathematical derivation of an optimal scheduling policy. Off-line scheduling can be formulated into an optimiza-

tion problem, and its solution is an optimal scheduling policy. According to QoS requirements, scheduling policies have different forms and the optimal scheduling policy for combined QoS requirements can be derived. The reason we solve an off-line scheduling scheme is that its scheduling policies are also optimal in an opportunistic scheduling scheme. Although the definitions of used terms are slightly different, opportunistic scheduling policies have analogous form with off-line scheduling policies. The optimalities of opportunistic scheduling policies can be proved. The opportunistic scheduling scheme is formulated into a stochastic optimization problem, and scheduling parameter updating algorithms are proposed by using a stochastic approximation. Opportunistic scheduling does not use unrealistic assumptions.

A. *Optimal Scheduling Policy Derivation with the Scheduling Interval of N Slots*

The purpose of an off-line optimal scheduling policy is to facilitate the conjecture of optimal scheduling policies according to QoS requirements. An off-line scheduling policy considers a scheduling interval which we set at N slots. We assume that the channel rate of user m at slot n, r_{mn}, is known when scheduled. Actually the assumption is unrealistic. In a real system, r_{mn}s for all n can not be known because MSes just feed back the channel status of the previous slot. To formulate scheduling as an optimization problem, the assumption is necessary so let's admit it. The scheduler determines the transmission sequence every N slots. The objective of the scheduling is the maximization of utility. We express the utility function for user m as

$$U_m = U_m\left(\frac{1}{N}\sum_{n=1}^{N}\rho_{mn}r_{mn}\right) \tag{1}$$

where ρ_{mn} indicates a portion of slot n allocated for user m. The utility function is assumed to be monotonically increasing with the average throughput. It is concave and differentiable. We represent its first order derivative as $U'_m(\cdot)$. Since only a single user is assigned in a slot, $\rho_{mn} = 1$ if user m is scheduled in slot n and $\rho_{mn} = 0$ in other case. If we define s_n as the scheduled user at slot n, the off-line scheduling becomes finding the vector $S = (s_1, \cdots, s_N)$ having the maximum utility value in $N \times M$ vectors. Though it needs much calculation, it is not hard to find optimal vector S^*. Instead we focus on finding the property of optimal S^* because from the property we can derive the simple scheduling policy.

Referring to the utility maximization as an unconstrained optimum problem, we formulate it as follows. Contrary to an original problem where only one user is served at each slot, we allow several users to share a slot by dividing it into smaller pieces. This relaxation makes the theoretical analysis simple.

$$\begin{aligned} &\text{maximize} \sum_{m=1}^{M} U_m(r_m) \\ &\text{subject to} \sum_{m=1}^{M}\rho_{mn} = 1, \\ &\qquad 0 \le \rho_{mn} \le 1, \\ &\qquad \text{for } n = 1, \cdots, N \text{and } m = 1, \cdots, M \end{aligned} \tag{2}$$

where $r_m (= 1/N \sum_{n=1}^{N} \rho_{mn} r_{mn})$ is the average service rate of user m for the period of N. This is a convex optimization problem because the objective function is concave and the feasible set is convex, so it yields a unique solution.

Lemma 1: The total utility is maximized by serving user m_n^*, at slot n where

$$\begin{aligned} m_n^* &= \arg\max_m U'_m(r_m^*)\, r_{mn}, \\ r_m^* &= 1/N \sum_{n=1}^{N} \rho_{mn}^* r_{mn}. \end{aligned} \tag{3}$$

r_m^* is determined after all slots are scheduled, so the equation is expressed in recursive form. We omit the proof of Lemma 1 because it is a particular case of Theorem 1 that will be given in Appendix A. Lemma 1 does not hold if more than a user has the same $U'_m(r_m^*)\, r_{mn}$ at slot n. A non-integer value of ρ_{mn} leads to an optimal solution but it is not achievable in a practical time division multiplexing system. So, we need a tie-breaking rule like random selection which determines a non-integer value of ρ_{mn} when $U'_m(r_m^*)\, r_{mn}$'s of more than one user are the same.

To make the scheduling policy achieve not only generalized proportional fairness but also guarantee QoS, we consider three types of QoS constraints for the utility maximization problem: Temporal share constraint, minimum throughput constraint, and throughput-share constraint. For the temporal share constraint, we can express the constraint for user m as

$$\frac{1}{N}\sum_{n=1}^{N}\rho_{mn} \ge a_m, \forall\, m \tag{4}$$

where a_m is the minimum portion of time slots that need to be allocated for user m, and a_m should satisfy $\sum_{m=1}^{M} a_m \le 1$. If a new comer cannot be admitted because time slots are lacking, a negotiation process may follow.

Lemma 2: For the temporal share constraint problem, the total utility is maximized by serving user m_n^* at slot n, where

$$m_n^* = \arg\max_m \{U'_m(r_m^*)\, r_{mn} + \lambda_m\} \tag{5}$$

and λ_m is the Lagrange multiplier for user m.

The proof of Lemma 2 also can be included in the proof of Theorem 1. In Lemma 2, we use a Lagrange multiplier to achieve the optimality. The Lagrange multiplier works as a compensation factor to meet each user's QoS requirements that could not be met in standard PF scheduling. If user m has a good channel, its QoS can be met by setting λ_m at zero. Otherwise, λ_m should have a positive value. A detailed algorithm to obtain λ_m will be given later.

The minimum throughput and throughput-share requirements for user m can be written as follows:

$$r_m \ge b_m, \tag{6}$$

$$r_m \ge c_m \sum_{i=1}^{M} r_i, \forall\, m \tag{7}$$

where b_m is the minimum throughput required by user m and

c_m is a portion of the total throughput required by user m. From the throughput-share constraint, we have $\sum_{i=1}^{M} c_i \leq 1$. For the minimum throughput constraint, no exact condition exists because the total system capacity is not fixed and depends upon users' channel conditions and the scheduling result. If the scheduling result is unstable for guaranteeing each user's minimum throughput, a negotiation process should request relevant users to relax their minimum throughput requirements.

Lemma 3: For the minimum throughput constraint problem, the total utility is maximized by serving user m_n^* at slot n where

$$m_n^* = \arg\max_m \left\{U_m'\left(r_m^*\right) + \mu_m\right\} r_{mn}. \tag{8}$$

Lemma 4: For the throughput-share constraint problem, the total utility is maximized by serving user m_n^* at slot n where

$$m_n^* = \arg\max_m \left\{U_m'\left(r_m^*\right) + \phi_m - \pi\right\} r_{mn} \tag{9}$$

and $\pi = \sum_{m=1}^{M} \phi_m c_m$.

The constants μ_m's and ϕ_m's are the corresponding Lagrange multipliers, respectively. We can prove Lemmas 3 and 4 by using the proof of Theorem 1, so we omitted them. Lemmas 2 through 4 do not hold if more than one user has the largest value of scheduling equation at some slot, so a tie-breaking rule is necessary again. These problems with added QoS constraints still maintain the form of convex optimization.

There is a possibility that each user has different QoS requirements. That is, one user requires temporal share performance while other users require minimum throughput performance. Some users may require temporal share and throughput-share performances together. This motivates us to consider a scheduling for combined QoS requirements. The scheduling can be formulated as an optimization problem as follows:

$$\begin{aligned}
&\text{maximize} \sum_{m=1}^{M} U_m\left(r_m\right) \\
&\text{subject to} \sum_{m=1}^{M} \rho_{mn} = 1, \\
&\quad 0 \leq \rho_{mn} \leq 1, \\
&\quad \frac{1}{N}\sum_{n=1}^{N} \rho_{mn} \geq a_m, \forall m \\
&\quad r_m \geq b_m, \forall m \\
&\quad r_m \geq c_m \sum_{i=1}^{M} r_i, \forall m \\
&\quad \text{for } n = 1, \cdots, N, \text{and } m = 1, \cdots, M
\end{aligned} \tag{10}$$

where $r_m (= 1/N \sum_{n=1}^{N} \rho_{mn} r_{mn})$.

Theorem 1: For the combined constraint problem, the total utility is maximized by serving user m_n^* at slot n, where

$$m_n^* = \arg\max_m \left\{U_m'\left(r_m^*\right) + \mu_m + \phi_m - \pi\right\} r_{mn} + \lambda_m \tag{11}$$

and $\pi = \sum_{m=1}^{M} \phi_m c_m$. The non-negative parameters λ_m's, μ_m's, and ϕ_m's satisfy the conditions of $\lambda_m(-\frac{1}{N}\sum_{n=1}^{N} \rho_{mn} + a_m) = 0$, $\mu_m(r_m - b_m) = 0$, and $\phi_m(r_m - c_m \sum_{i=1}^{M} r_i) = 0$.

The proof of Theorem 1 is given in Appendix A. In the equation, Lagrange multipliers are independently determined and the tie-breaking problem still remains.

B. Opportunistic Optimal Scheduling Policy

We consider the case of utility maximization that does not have the scheduling interval constraint. Unlike off-line scheduling, this scheduling does not use the assumption of knowing the channel rates of several slots beforehand. Let's assume that the scheduler knows each user's channel rate for the current slot. Its policy considers the total utility maximization and QoS guarantee for each user in a probabilistic manner. We represent the instantaneous transmission rate for user m at slot n as R_{mn} that is similar to r_{mn} in the case of scheduling interval of N slots, but R_{mn} is a random variable while r_{mn} is constant. Assume that R_{mn} is a stationary process. Then, we can remove its slot index without ambiguity, and user m transmits at a rate of R_m when scheduled. The scheduling policy Q takes the feasible rate of each user as input and chooses a user to be served. It gives the average throughput of $\overline{R}_m^Q = E(R_m I_{\{Q=m\}})$ for user m, where the indicator function $I_{\{Q=m\}}$ is 1 if user m is selected, and 0 otherwise.

Denoting the utility function for user m as $U_m = U_m(\overline{R}_m^Q)$, we can formulate this problem as follows:

$$\max_Q \sum_{m=1}^{M} U_m\left(\overline{R}_m^Q\right). \tag{12}$$

The following theorem describes a form of optimal scheduler that satisfies the above objective.

Lemma 5: If the scheduling interval is not given, the scheduling policy

$$Q^* = \arg\max_m U_m'\left(\overline{R}_m^{Q^*}\right) R_m \tag{13}$$

maximizes the total utility.

Proof: For any feasible scheduling policy Q, we have

$$\sum_{m=1}^{M} U_m'\left(\overline{R}_m^{Q^*}\right) R_m I_{\{Q=m\}} \leq \sum_{m=1}^{M} U_m'\left(\overline{R}_m^{Q^*}\right) R_m I_{\{Q^*=m\}} \tag{14}$$

where $\overline{R}_m^{Q^*}$ is the user m's average throughput obtainable by scheduler Q^*. Considering the expectation on both sides, we obtain

$$\begin{aligned}
&\sum_{m=1}^{M} U_m'\left(\overline{R}_m^{Q^*}\right) \overline{R}_m^Q \leq \sum_{m=1}^{M} U_m'\left(\overline{R}_m^{Q^*}\right) \overline{R}_m^{Q^*} \\
&\qquad \Leftrightarrow \sum_{m=1}^{M} U_m'\left(\overline{R}_m^{Q^*}\right)\left(\overline{R}_m^Q - \overline{R}_m^{Q^*}\right) \leq 0.
\end{aligned} \tag{15}$$

Using the vector notations, we get

$$\nabla U(\overline{\mathbf{R}}_{Q^*})(\overline{\mathbf{R}}_Q - \overline{\mathbf{R}}_{Q^*}) \leq 0 \tag{16}$$

where $\nabla U(\overline{\mathbf{R}}) = (\mathbf{U_1'}(\overline{\mathbf{R}}_\mathbf{1}), \cdots, \mathbf{U_M'}(\overline{\mathbf{R}}_\mathbf{M}))$, $\overline{\mathbf{R}}_Q = (\overline{R}_1^Q, \cdots, \overline{R}_M^Q)$, and $\overline{\mathbf{R}}_{Q^*} = (\overline{R}_1^{Q^*}, \cdots, \overline{R}_M^{Q^*})$. This means that $U(\overline{\mathbf{R}})$ has the maximum at $\overline{\mathbf{R}} = \overline{\mathbf{R}}_{\mathbf{Q}^*}$. □

The scheduling policy in Lemma 5 is very similar to that in Lemma 1. The differences are as follows: i) Whether the scheduling policy is off-line or on-line and ii) whether the channel rate is deterministic or stochastic. There is no need of a tie-breaking rule in the opportunistic scheduling because it does not influence the scheduling result in the long term average. Even when a tie-breaking occurs, the effect of temporarily unbalanced allocation does not last long owing to the compensation of later schedulings. Therefore our opportunistic scheduling policy results in a stochastic global optimum. In subsection II-D, we deal with the convergence of the proposed scheduling policy.

The scheduling policy given in Lemma 5 becomes HDR scheduler,[1] which is one of the implemented PF schedulers, if the utility function has a logarithmic form of the throughput rate.

$$Q^* = \arg\max_m U'_m\left(\overline{R}_m^{Q^*}\right) R_m = \arg\max_m \frac{R_m}{\overline{R}_m^{Q^*}}. \quad (17)$$

This scheduling policy selects a user that has a normalized maximum rate. Obviously it yields the maximum throughput if the utility is a linear function of throughput.

Now we develop opportunistic schedulers that maximize the total utility while satisfying each user's QoS requirements. According to the previous three constraints, we can obtain the following three facts, respectively. First, we consider the case that user m requires a slot with the minimum probability α_m.

Lemma 6: The scheduling policy

$$Q^* = \arg\max_m \left\{U'_m\left(\overline{R}_m^{Q^*}\right) R_m + \lambda_m^*\right\} \quad (18)$$

maximizes the total utility under the constraint $\Pr\{Q^* = m\} \geq \alpha_m$, where the non-negative parameters λ_m^*'s satisfy the condition $\lambda_m^*(\Pr\{Q^* = m\} - \alpha_m) = 0$.

The optimality of this opportunistic scheduling is achieved by the utility maximization and the QoS guarantee in addition to the adaptation of λ_m. A user in bad channel needs an appropriate λ_m to receive the desired QoS in the utility maximization policy. To compensate for the QoS gap, λ_m should have a positive value, so the utility maximization problem is constrained by λ_m's.

Second, we consider the case that user m requires the minimum average throughput of β_m.

Lemma 7: The scheduling policy

$$Q^* = \arg\max_m \left\{U'_m\left(\overline{R}_m^{Q^*}\right) + \mu_m^*\right\} R_m \quad (19)$$

maximizes the total utility under the constraint $\overline{R}_m^{Q^*} \geq \beta_m$, where the non-negative parameters μ_m^*'s satisfy the condition $\mu_m^*(\overline{R}_m^{Q^*} - \beta_m) = 0$.

Lastly, we consider the case that user m requires a portion γ_m of the total throughput.

Lemma 8: The scheduling policy

$$Q^* = \arg\max_m \left\{U'_m\left(\overline{R}_m^{Q^*}\right) + \phi_m^* - \pi\right\} R_m \quad (20)$$

[1]HDR scheduler uses a sliding window algorithm to calculate average throughput.

maximizes the total utility under the constraint $\overline{R}_m^{Q^*} \geq \gamma_m \sum_{i=1}^M \overline{R}_i^{Q^*}$ where $\pi = \sum_{m=1}^M \phi_m^* \gamma_m$, and the non-negative parameters ϕ_m^*'s satisfy the condition $\phi_m^*(\overline{R}_m^{Q^*} - \gamma_m \sum_{i=1}^M \overline{R}_i^{Q^*}) = 0$.

We omit the proofs of Lemmas from 6 to 8 because they are similar to that for the following combined scheduling policy that can support each user's heterogeneous and various QoS requirements. Like before, there exist μ's and ϕ's lead the scheduling policies to meet the minimum throughput and throughput-share requirements.

If each user has different QoS requirements in opportunistic scheduling, the following scheduling policy can be used.

Theorem 2: The scheduling policy

$$Q^* = \arg\max_m \left\{U'_m\left(\overline{R}_m^{Q^*}\right) + \mu_m^* + \phi_m^* - \pi\right\} R_m + \lambda_m^* \quad (21)$$

maximizes the total utility under the constraints $\Pr\{Q^* = m\} \geq \alpha_m$, $\overline{R}_m^{Q^*} \geq \beta_m$, and $\overline{R}_m^{Q^*} \geq \gamma_m \sum_{i=1}^M \overline{R}_i^{Q^*}$ where $\pi = \sum_{m=1}^M \phi_m^* \gamma_m$. The non-negative parameters λ_m^*'s, μ_m^*'s, and ϕ_m^*'s satisfy the conditions of $\lambda_m^*(\Pr\{Q^* = m\} - \alpha_m) = 0$, $\mu_m^*(\overline{R}_m^{Q^*} - \beta_m) = 0$ and $\phi_m^*(\overline{R}_m^{Q^*} - \gamma_m \sum_{i=1}^M \overline{R}_i^{Q^*}) = 0$.

This policy successfully satisfies all the three QoS requirements and its optimality is proved in Appendix B; therefore, it can support any type of QoS. For example, if user m requires the temporal share of 10%, the QoS parameters are set at $\alpha = 0.1$, $\beta = 0$, and $\gamma = 0$. If user l requires a temporal share of 20% and a minimum throughput of 50 kbps, the QoS parameters are set at $\alpha = 0.2$, $\beta = 50,000$, and $\gamma = 0$. For parameter adaptation, λ, μ, and ϕ are computed in accordance with the channel conditions and QoS requirements of all the users.

C. Parameter Adaptation Algorithm

In implementing optimal schedulers, we need to find Lagrange multipliers and calculate the average throughput. For Lagrange multipliers, a parameter adaptation algorithm is needed to adapt those estimated values to optimal ones. The stochastic approximation theory plays an important role in ensuring the estimated values to approach the optimal ones if QoS requirements are met. This means that the scheduling result becomes stable and optimal. Our proposed algorithm considers the temporal share constraint only, but it can be easily applied for other constraints also.

Lagrange multipliers from Lemma 5 must satisfy the following optimality conditions.

$$\begin{aligned} \lambda_m^* &\geq 0 \,(\text{non-negativity}), \\ \Pr\{Q^* = m\} - \alpha_m &\geq 0 \,(\text{feasibility}), \\ \lambda_m^*\left(\Pr\{Q^* = m\} - \alpha_m\right) &= 0 \,(\text{complementary slackness}). \end{aligned} \quad (22)$$

The non-negativity condition is a property of the Lagrange multiplier, and the feasibility condition comes from the QoS constraint that user m's temporal share should be equal to or greater than α_m. The complementary slackness describes the condition for a point to be optimal and stable where a user that receives a temporal share greater than its requirement has zero λ. In contrast, a user that does not receive as large of a temporal share as

Table 1. Noisy observations.

QoS requirement	Noisy observaton
Temporal share	$g_{m,k} = I_{\{Q_k=m\}} - \alpha_m$
Minimum throughput	$g_{m,k} = R_m I_{\{Q_k=m\}} - \beta_m$
Throughput-share	$g_{m,k} = R_m I_{\{Q_k=m\}} - \gamma_m \sum_{i=1}^{M} R_i I_{\{Q_k=i\}}$

required has a non-negative λ that is used to overcome the utility value due to poor channel conditions in providing required QoS.

Since these conditions do not specify what λ's should be, our parameter adaptation algorithm aims at finding those values by a stochastic approximation. To do so, we define a function

$$f_m^0(\lambda_m) = \lambda_m (Pr\{Q = m\} - \alpha_m), \quad \text{for } m = 1, \cdots, M \tag{23}$$

and search for its zeros on the non-negative region. Actually, $f_m^0(\lambda_m)$ has a trivial zero of $\lambda_m = 0$, so we need to search for zeros of the function

$$f_m(\lambda_m) = Pr\{Q = m\} - \alpha_m, \quad \text{for } m = 1, \cdots, M. \tag{24}$$

In the kth iteration (or slot), we use estimated $\lambda_{m,k}$ and select a user to be served by an intermediately obtained scheduler Q_k. Here, Q_k is not an optimal scheduler because it uses transient $\lambda_{m,k}$'s instead of λ_m^*'s. Also, $f_m(\lambda_{m,k})$ contains a probability term that makes its accurate value unattainable. By introducing a noisy observation

$$g_{m,k} = I_{\{Q_k=m\}} - \alpha_m \tag{25}$$

we can solve the parameter adaptation problem. That is

$$\lambda_{m,k+1} = \max(\lambda_{m,k} - \delta_k g_{m,k}, 0) \tag{26}$$

where δ_k is a step sequence for adaptation. The max function implements the projection onto the non-negative region, and the step sequence satisfies the following: $\delta_k > 0$, $\delta_k \to 0$, and $\sum_k \delta_k \to \infty$. The function $f_m(\lambda_m)$ increases with λ_m in a monotonic manner so that it has a unique zero, which we denote as z. If $z < 0$, $\lambda_{m,k}$ converges z and it is forced to approach zero. From $f_m(0) > f_m(z) = 0$, we can find that λ_m of zero satisfies all the three conditions. If $z > 0$, $\lambda_{m,k}$ converges at z, and it becomes optimal since $f_m(z) = 0$. From these observations, we can conclude that our algorithm always gives a correct answer.

The noisy observations for minimum throughput and throughput-share requirements can be obtained in a similar way. Table 1 summarizes these results.

D. Convergence of Scheduling Algorithms

The proposed scheduling system has the property of optimality and convergence. In this subsection, we deal with the convergence of the parameter adaptation part. In combined QoS scheduling, there are four parameters that need updating at every slot. These are average throughput $\overline{\mathbf{R}}$ and QoS parameters λ, μ, and ϕ. The average throughput is updated by

$$\overline{R}_{m,k+1} = \overline{R}_{m,k} + \varepsilon_k \left[R_{m,k+1} I_{m,k+1} - \overline{R}_{m,k}\right] \tag{27}$$

where $\varepsilon_k = 1/k$ and $I_{m,k}$ is an indicator function that equals 1 if the scheduler selects MS m at slot k, otherwise 0. Then, we obtain $\overline{R}_{m,k} = 1/k \sum_{i=1}^{k} R_{m,i} I_{m,i}$. The updating algorithms for the QoS parameters are introduced in subsection II-C. The analysis procedures are given in [23] and [15].

Before considering the convergence of throughput, we define the shifted process $\overline{R}_{m,k}$ as the follows.

$$\overline{R}_{m,k}(t) = \overline{R}_{m,k+l} \text{ for } t \in \left[\sum_{j=k}^{k+l-1} \varepsilon_j, \sum_{j=k}^{k+l} \varepsilon_j\right) \tag{28}$$

$\overline{R}_{m,k}(t)$ is differentiable, and $\overline{R}_{m,k}(t)$ and the original sequence $\overline{R}_{m,k}$ show the same behavior as k goes towards infinity. Under the assumption that instantaneous R_m is stationary, we can define its expectation as follows according to [15]:

$$\overline{h}_m(\overline{\mathbf{R}}) = E\left[R_m I_{\left\{F_m(\overline{\mathbf{R}};\lambda,\mu,\phi) \geq F_j(\overline{\mathbf{R}};\lambda,\mu,\phi), j \neq m\right\}}\right] \tag{29}$$

where $F_m(\overline{\mathbf{R}}, \lambda, \mu, \phi) = \{U'(\overline{R}_m) + \mu_m + \phi_m - \pi\} R_m + \lambda_m$ and $F_m(\overline{\mathbf{R}}; \lambda, \mu, \phi)$ is a function of $\overline{\mathbf{R}}$ when λ, μ, and ϕ are given.

Theorem 3: If the initial condition is given as the origin vector, $\overline{\mathbf{R}}$ weakly converges to the unique solution of the following ordinary differential equation (ODE)

$$\dot{\overline{R}}_m = \overline{h}_m(\overline{\mathbf{R}}) - \overline{R}_m. \tag{30}$$

Proof: According to [15], this theorem can be sufficiently proved by showing that $\mathbf{f}(\overline{\mathbf{R}})$ satisfies the Kamke condition (K-condition) where $\mathbf{f}(\overline{\mathbf{R}})$ is defined as

$$\mathbf{f}(\overline{\mathbf{R}}) = \overline{\mathbf{h}}(\overline{\mathbf{R}}) - \overline{\mathbf{R}}. \tag{31}$$

If $\mathbf{f}(\overline{\mathbf{R}})$ satisfies $f_m(\hat{\overline{\mathbf{R}}}) \leq f_m(\tilde{\overline{\mathbf{R}}})$ when $\hat{\overline{\mathbf{R}}} \leq \tilde{\overline{\mathbf{R}}}$ and $\hat{\overline{R}}_m = \tilde{\overline{R}}_m$, f satisfies K-condition. For vector inequality in K-condition, we define $\hat{\overline{\mathbf{R}}} \leq \tilde{\overline{\mathbf{R}}}$ when $\hat{\overline{R}}_j \leq \tilde{\overline{R}}_j$ for all j. Since $U_j(\overline{\mathbf{R}})$ is concave, $U_j'(\hat{\overline{R}}_j) \geq U_j'(\tilde{\overline{R}}_j)$ if $\hat{\overline{R}}_j \leq \tilde{\overline{R}}_j$. For this reason, if $\hat{\overline{R}}_j \leq \tilde{\overline{R}}_j$, we have

$$F_j\left(\hat{\overline{\mathbf{R}}}; \lambda, \mu, \phi\right) \begin{cases} \geq F_j\left(\tilde{\overline{\mathbf{R}}}; \lambda, \mu, \phi\right), & j \neq m, \\ = F_j\left(\tilde{\overline{\mathbf{R}}}; \lambda, \mu, \phi\right), & j = m. \end{cases} \tag{32}$$

This results in the following relation.

$$\begin{aligned} f_m\left(\hat{\overline{\mathbf{R}}}\right) &= E\left[R_m I_{\left\{F_m\left(\hat{\overline{\mathbf{R}}};\lambda,\mu,\phi\right) \geq F_j\left(\hat{\overline{\mathbf{R}}};\lambda,\mu,\phi\right), j \neq m\right\}}\right] - \hat{\overline{R}}_m \\ &\leq E\left[R_m I_{\left\{F_m\left(\tilde{\overline{\mathbf{R}}};\lambda,\mu,\phi\right) \geq F_j\left(\tilde{\overline{\mathbf{R}}};\lambda,\mu,\phi\right), j \neq m\right\}}\right] - \tilde{\overline{R}}_m \\ &= f_m\left(\tilde{\overline{\mathbf{R}}}\right). \end{aligned} \tag{33}$$

So, $\mathbf{f}(\overline{\mathbf{R}})$ satisfies K-condition. □

Theorem 3 guarantees the convergence of throughput when QoS parameters are given. Since QoS parameters are also updated at every slot by the parameter adaptation procedures, the convergence behavior of QoS parameter updating algorithms

should be verified. The convergence of the algorithms can be shown through the similar procedures as in Theorem 3 and the definition of ODE and K-condition. The following lemmas deal with the convergence of QoS parameters.

Lemma 9: When the initial condition is given as the origin vector, λ converges weakly to the unique solution of the following ODE

$$\dot{\lambda}_m = \overline{g}_m^{\lambda}(\lambda). \tag{34}$$

Lemma 10: When the initial condition is given as the origin vector, μ converges weakly to the unique solution of the following ODE

$$\dot{\mu}_m = \overline{g}_m^{\mu}(\mu). \tag{35}$$

Lemma 11: When the initial condition is given as the origin vector, ϕ converges weakly to the unique solution of the following ODE

$$\dot{\phi}_m = \overline{g}_m^{\phi}(\phi). \tag{36}$$

In Lemma 9, we define $\overline{g}_m^{\lambda}(\lambda)$ as

$$\overline{g}_m^{\lambda}(\lambda) = E\left[\alpha_m - I_{\left\{F_m\left(\lambda;\overline{\mathbf{R}},\mu,\phi\right)\geq F_j\left(\lambda;\overline{\mathbf{R}},\mu,\phi\right),j\neq m\right\}}\right]. \tag{37}$$

Equation (37) comes from the noisy observation (25) in the parameter adaptation algorithm. Similarly, $\overline{g}_m^{\mu}(\mu)$ and $\overline{g}_m^{\phi}(\phi)$ are defined as

$$\overline{g}_m^{\mu}(\mu) = E\left[\beta_m - R_m I_{\left\{F_m\left(\mu;\overline{\mathbf{R}},\lambda,\phi\right)\geq F_j\left(\mu;\overline{\mathbf{R}},\lambda,\phi\right),j\neq m\right\}}\right], \tag{38}$$

$$\overline{g}_m^{\phi}(\phi) = E\Big[\gamma_m \sum_{i=1}^{M} R_i I_{\left\{F_i\left(\phi;\overline{\mathbf{R}},\lambda,\mu\right)\geq F_j\left(\phi;\overline{\mathbf{R}},\lambda,\mu\right),j\neq i\right\}} - R_m I_{\left\{F_m\left(\phi;\overline{\mathbf{R}},\lambda,\mu\right)\geq F_j\left(\phi;\overline{\mathbf{R}},\lambda,\mu\right),j\neq m\right\}}\Big], \tag{39}$$

respectively. The proof of Lemma 9 is given in Appendix C. Lemmas 10 and 11 can be proved similarly. As these parameters are updated simultaneously, their convergences should be guaranteed. In the proof, other parameters (such as λ, μ, and ϕ in Theorem 3) are assumed to be fixed, so the results have a limited meaning.

III. SIMULATION RESULTS

A. Performance of Optimal Scheduling Policies

We performed simulations for optimal scheduling policies under the requirements of temporal share (scenario 1), minimum throughput (scenario 2), and throughput-share (scenario 3). Scenario 4 deals with a heterogeneous requirement case. In scenario 5, the proposed scheduling algorithm is compared to other scheduling algorithms. We assume that physical channel characteristics can be abstracted to a user's feasible data rate performance according to a probabilistic model, so we consider five users whose feasible rates are exponentially distributed with the mean of 100, 200, 300, 400, and 500 (kbps), respectively in scenario 1, 2, 3, and 5. In scenario 4, we consider 20 users that have good, medium, and bad channel conditions. A detailed channel description of scenario 4 is given in the result explanation part.

Table 2. Temporal share and utility for scenario 1.

User	Case 1 (10%)		Case 2 (20%)	
	Temporal share	Utility	Temporal share	Utility
1	10.08	19.42	19.69	32.15
2	10.52	55.28	19.49	84.45
3	16.92	122.77	19.42	138.14
4	26.56	223.51	19.87	193.86
5	35.92	339.73	21.53	260.99
Total	100.0	760.71	100.0	709.59

Table 3. Average throughput and utility for scenario 2.

User	Case 1 (0 kbps)		Case 2 (50 kbps)	
	Avg. throughput	Utility	Avg. throughput	Utility
1	45.85	3.83	52.56	3.96
2	91.21	4.51	71.17	4.26
3	136.10	4.91	106.35	4.67
4	184.20	5.22	148.34	5.00
5	229.74	5.44	192.18	5.26
Total	687.10	23.91	570.60	23.15

Scenario 1 has the utility function of $U_m\left(R_m^Q\right) = R_m^Q$, and scenarios 2 through 4 have $U_m\left(R_m^Q\right) = \ln R_m^Q$. For the parameter adaptation algorithm, the step sequence is given as $1000/(k+1)$ for scenario 1, and $1/(k+1)$ for scenarios 2 and 3. Simulations are executed for the duration of 10,000 slots and the scheduling parameters are initially set to zero.

We consider the minimum temporal share requirements of 10% (case 1) and 20% (case 2) in scenario 1. As the utility function is linear, the scheduler targets maximizing the throughput. The results are summarized in Table 2. The scheduler strictly meets the temporal share requirements in the both cases. In case 1, the scheduler allocates a minimum number of slots for users 1 and 2 because their channels are poor. To maximize the utility, the scheduler prefers users in good channels, thereby allocating more slots for them. The results indicate that user 5 has the best channel. In case 2, slots are allocated according to the users' temporal share requirements as the slot utilization reaches 100%. The total utility in case 1 is larger than that in case 2 because case 1 has more flexibility in scheduling.

In scenario 2, we set the minimum throughput requirement at 0 kbps (case 1) and 50 kbps (case 2), respectively. Our scheduler satisfies the minimum throughput requirement for each user successfully, as shown in Table 3. Case 1 has no constraint, so the scheduler acts like a PF scheduler because the utility function has a logarithmic form. In case 2, there is a minimum throughput requirement, so the total utility in case 2 is less than that in case 1. This is because the gain in case 2 is limited by the constraint.

In scenario 3, we set the throughput-share requirement at 10% (case 1) and 20% (case 2), respectively. Table 4 shows that our scheduler satisfies the throughput-share requirement for each user successfully. Like in scenario 1, case 1 confirms that our scheduler works well, even when some users experience bad channels. After meeting each user's requirement, it allocates the remaining slots for users in good channels. In case 2, the sum of throughput share requirements is 100%. As the requirement

Table 4. Throughput-share and utility for scenario 3.

User	Case 1 (10%)		Case 2 (20%)	
	Throughput-share	Utility	Throughput-share	Utility
1	11.86	4.00	20.32	4.23
2	14.65	4.22	20.05	4.22
3	19.53	4.50	19.96	4.21
4	24.68	4.73	19.89	4.21
5	29.28	4.90	19.78	4.21
Total	100.0	22.35	100.0	21.08

Table 5. Heterogeneous QoS requirements and scheduling results for scenario 4.

User	Temporal share		Average throughput		Throughput share	
	Req. (%)	Result (%)	Req. (kbps)	Result (kbps)	Req. (%)	Result (%)
1	2	2.12	0	12.94	0	0.71
2	0	2.58	20	20.71	0	1.13
3	0	5.54	0	37.17	2	2.04
4	1	2.63	20	20.64	0	1.13
5	1	2.32	0	18.75	1	1.03
6	0	2.67	20	20.71	1	1.13
7	1	2.70	20	20.32	1	1.11
8	0	2.62	0	25.42	0	1.39
9	1	2.04	20	24.12	0	1.32
10	1	3.67	0	39.33	2	2.15
11	1	1.99	20	23.07	0	1.26
12	0	1.74	0	21.93	1	1.20
13	1	2.26	20	26.21	1	1.44
14	2	2.23	20	26.03	1	1.43
15	0	9.37	0	224.60	0	12.30
16	0	9.91	20	237.63	0	13.01
17	1	11.34	0	267.74	2	14.66
18	1	10.54	20	250.31	0	13.71
19	1	10.66	0	255.98	1	14.02
20	0	11.07	20	252.62	1	13.83
Total		100		1826.22		100

sum is too tight, too much room exits for scheduling, so the total utility becomes smaller than in case 1.

In scenario 4, we examined a more realistic situation. There are 20 users, and they have different channel states and QoS requirements. There are three channel states in this scenario. Users 1 through 7 have bad channels, so their feasible rates are exponentially distributed with a mean of 100. Users 8 through 14 have medium channels, so their mean parameter is 200. Users 15 through 20 have 800 as mean parameters because they have good channels. The heterogeneous QoS requirements for each user and the simulation results are shown in Table 5. They confirm that our scheduler successfully meets the heterogeneous QoS requirements for each user under a realistic situation.

In scenario 5, we compare the proposed scheduling algorithm with the PF scheduling algorithm and weighted PF scheduling algorithm. We set the minimum throughput requirement at 70 kbps and 80 kbps for each user in cases 1 and 2, respectively. In the weighted PF scheduling algorithm, the weighting factor is multiplied by the PF scheduling metric. The weighting factors of user m are set by the algorithm in [24]. In [24], the weighting factor is determined by the required activity detection algorithm. Table 6 shows the throughput results of the scheduling algorithms. Proposed scheduling algorithm satisfies the minimum throughput requirement. The throughput performances of other scheduling algorithms are better, but they cannot guarantee the minimum throughput requirement.

Table 6. Throughput comparison for scenario 5.

User	PF	Case 1		Case 2	
		Prop.	W-PF	Prop.	W-PF
1	45.81	72.23	61.17	81.18	61.09
2	91.80	71.84	96.27	80.90	96.88
3	136.97	107.80	125.37	82.25	126.37
4	182.33	145.05	151.33	108.55	151.63
5	229.69	178.45	174.10	135.91	173.68
Total	686.60	575.36	608.24	488.79	609.65

B. Convergence of Optimal Scheduling Policies

To implement an optimal opportunistic scheduler in real systems, there are two basic requirements: simplicity and stability. In terms of complexity, our algorithm requires a few more additions for the parameter calculation compared to the HDR scheduler. For stability, our algorithm results in stable resource allocation and parameter values with fast convergence. To observe the convergence of our scheduler, we trace the received QoS level for each user and its Lagrange multipliers.

Assume that there are three users whose average feasible rates are 200, 300, and 800 (kbps), respectively. The utility function has a linear form. Figs. 2 and 3 show the convergence behaviors of our scheduler under the temporal share constraint of 20% for each user. The three users each receive each shares of 20.7, 20.7, and 58.6%, respectively. User 1 has the largest λ_1 of 130.1 because he/she has the worst channel, while user 2 has the smallest λ_2 of 0 because his/her channel condition is the best. All the parameters converge in about 3,000 slots at most, meaning that the QoS parameter of λ converges reasonably fast.

Figs. 4 and 5 show the convergence behaviors of our scheduler under the minimum throughput constraint of 100 (kbps). The number of users and their average feasible rates are the same as before. The utility function has a logarithmic form. Each user obtains the throughput of 113.1, 149.7, and 412.7 (kbps), respectively, which meet each user's QoS requirement well. The λ's of users 1, 2, and 3 converge at 0.0504, 0.0268, and 0.0108, respectively.

Figs. 6 and 7 show the convergence behaviors under the throughput-share constraint of 20% each. The other conditions are the same as those for the minimum throughput constraint. The throughput-shares of each user are 22.9%, 27.1%, and 50.0%, respectively.

Figs. 8 and 9 show the convergence behaviors of three users among 10 and 20 users under the temporal share constraints of 5% and 3%, respectively. Assume that the average feasible rates of user 1, user 2, and user 3 are 200, 300, and 800 (kbps), respectively, and the ratio of 200 and 800 (kbps) users among the total users is 30% and the ratio of 300 (kbps) users is 40%. The ratio of 200 and 800 (kbps) users is 30% and the ratio of 300 (kbps) users is 40%. The results confirm that our scheduler

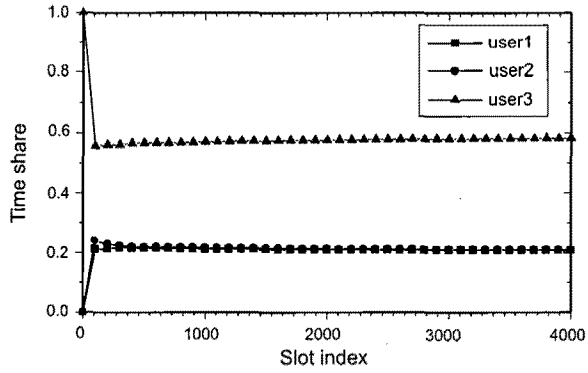

Fig. 2. Convergence of the temporal share between three users with 20% of the slot requirement each.

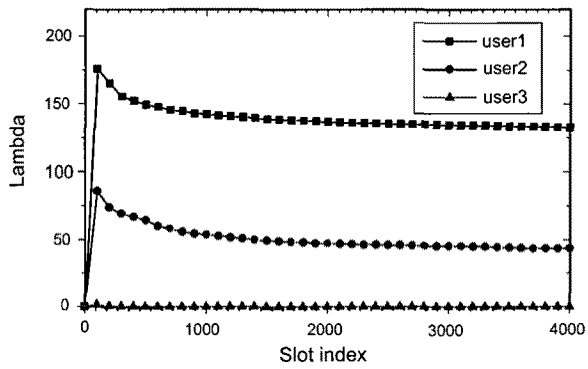

Fig. 3. Convergence of Lagrange multipliers for three users with 20% of the slot requirement each.

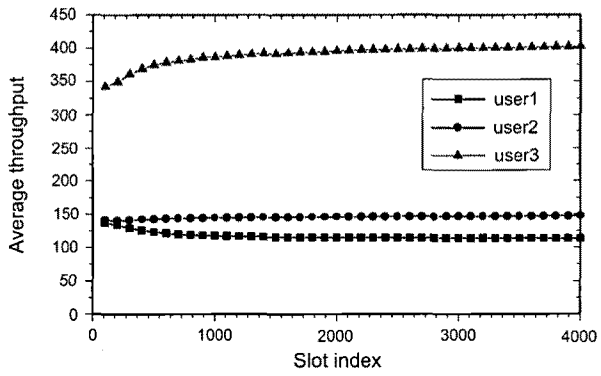

Fig. 4. Convergence of the average throughputs between three users with 150 kbps of the minimum throughput requirement each.

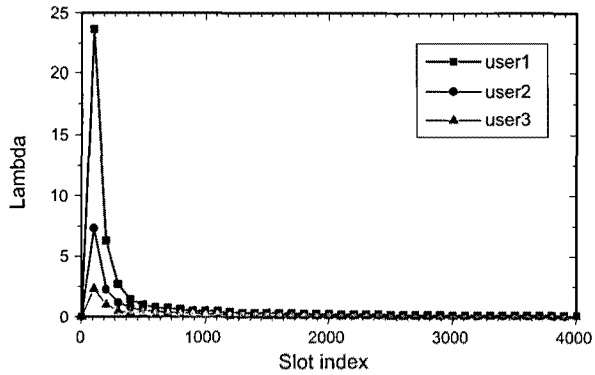

Fig. 5. Convergence of Lagrange multipliers for three users with 150 kbps of the minimum throughput requirement each.

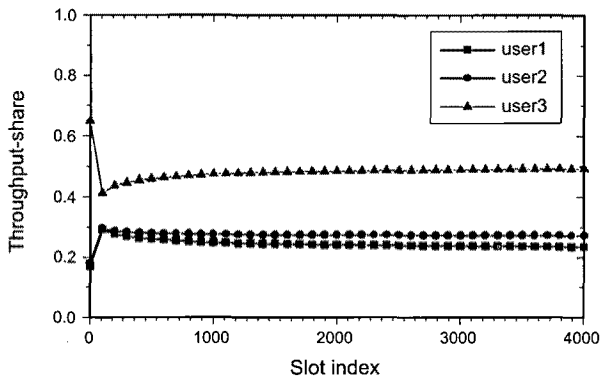

Fig. 6. Convergence of the throughput-share between three users with 20% of the throughput-share requirement each.

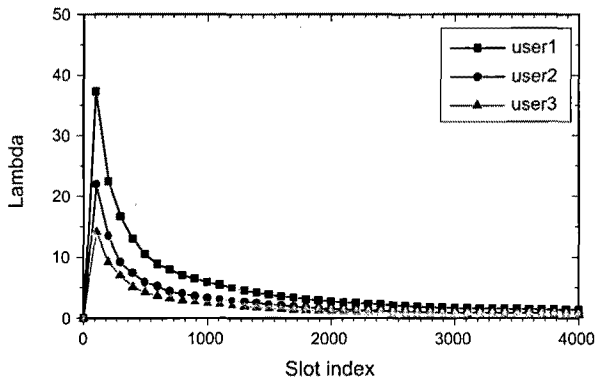

Fig. 7. Convergence of the Lagrange multipliers for three users with 20% of the throughput-share requirement each.

meets each user's QoS requirements successfully and the convergence speed is fast enough.

C. Discussion

From the simulation results, we observed that our scheduler maximizes the total utility and guarantees each user's QoS requirements, even when each user has different QoS requirements. Although our scheduler proved to be optimal, the optimal value does not converge if there are too many users in the system. Therefore, we need an admission control scheme that allows a limited number of users into the system, thereby guaranteeing each accepted user's QoS.

An advantage of our opportunistic scheduling is that its optimality is independent of the channel model. Only the mean and variance of the channel model affect the parameter adapta-

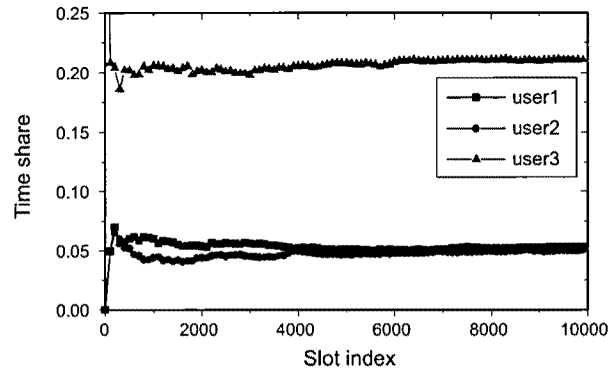

Fig. 8. Convergence of the temporal share between 10 users with 5% of the slot requirement each.

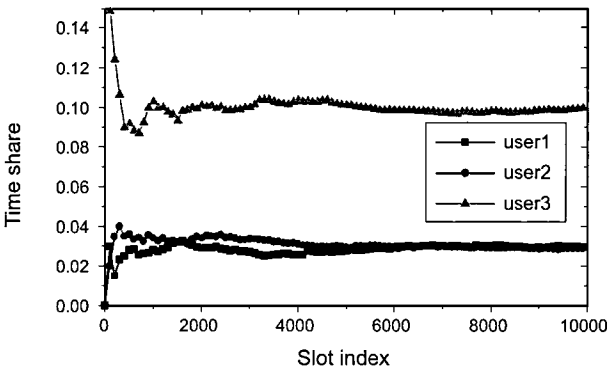

Fig. 9. Convergence of temporal share between 20 users with 3% of the slot requirement each.

tion. For the Gaussian channel model, we obtained similar results. The channel model affects the stability of opportunistic scheduling because the rate distribution for the scheduled slots depends on the channel model and determines the system capacity. If the sum of required resources exceeds the system capacity, the opportunistic scheduling becomes unstable.

In our scheduling algorithm, we selected a step sequence δ^k by trial and error that is the main factor in determining the speed of convergence. Finding an appropriate sequence is left for future work.

IV. CONCLUSION

In this paper, we proposed an optimal opportunistic scheduler that maximizes the total utility of a wireless system and meets each user's QoS requirements. Our considered QoS requirements are temporal share, minimum throughput, and throughput share. According to the considered scheduling interval, we considered two types of scheduling. One is the off-line scheduling where the assumption of knowing the channel rates in the scheduling interval beforehand is used. The other is the opportunistic scheduling, which considers the current slot information only. Interestingly, we obtained the same form of optimal schedulers for the two types.

We also developed an adaptive algorithm to find Lagrange multipliers for the optimal schedulers. It executes in an iterative manner and is not very complex. Through mathematic analysis we proved that the parameter updating algorithm gives convergent values. The speed of convergence can be made fast enough by some properly chosen step sequences. Through simulations, we confirmed that our schedulers work well while meeting each user's QoS requirements. The contributions of our work are as follows: 1) Deriving the optimal extended PF scheduling policies, 2) proving their optimalities considering QoS requirements, and 3) proving the convergence of updating algorithms. We need to work further to create an opportunistic scheduling system associated with admission control that guarantees each user's QoS requirements under heavy load.

APPENDIX

A. Proof of Theorem 1

Proof: The Lagrangian for (10) is given by

$$L = \sum_{m=1}^{M} U_m(r_m) - \sum_{n=1}^{N} \tau_n \left(\sum_{m=1}^{M} \rho_{mn} - 1 \right) - \sum_{m=1}^{M} \lambda_m \left(-\frac{1}{N} \sum_{n=1}^{N} \rho_{mn} + a_m \right) - \sum_{m=1}^{M} \mu_m (-r_m - b_m) - \sum_{m=1}^{M} \phi_m \left(-r_m + c_m \sum_{i=1}^{M} r_i \right) \tag{40}$$

where τ_n, λ'_m, μ_m, and ϕ_m are the Lagrange multipliers for slot n and user m, respectively. From Karuch-Kuhn-Tucker (KKT) conditions, for all n, we have

$$\frac{\partial L}{\partial \rho_{mn}} = \frac{1}{N} \left\{ \frac{dU_m(r_m)}{dr_m} r_{mn} - N\tau_n + \lambda_m + \mu_m r_{mn} + \phi_m r_{mn} - r_{mn} \sum_{i=1}^{M} \phi_i c_i \right\} \begin{cases} < 0, & \rho_{mn} = 0, \\ = 0, & 0 < \rho_{mn} < 1, \\ > 0, & \rho_{mn} = 1, \end{cases} \tag{41}$$

$$\sum_{m=1}^{M} \rho_{mn} - 1 = 0, \tag{42}$$

$$\lambda_m \left(-\frac{1}{N} \sum_{n=1}^{N} \rho_{mn} + a_m \right) = 0, \tag{43}$$

$$\mu_m \left(-\frac{1}{N} \sum_{n=1}^{N} \rho_{mn} r_{mn} + b_m \right) = 0, \tag{44}$$

$$\phi_m \left(-r_m + c_m \sum_{i=1}^{M} r_i \right) = 0, \tag{45}$$

$$\lambda_m \geq 0, \mu_m \geq 0, \phi_m \geq 0. \tag{46}$$

If a slot is assigned exclusively to a single user, we obtain the following relation from (41).

$$\left\{ U'_{m_n^*}(r_{m_n^*}) + \mu_m + \phi_m - \pi \right\} r_{m_n^* n} + \lambda_m > N\tau_n > \{ U'_m(r_m) + \mu_m + \phi_m - \pi \} r_{mn} + \lambda_m, \forall m \neq m_n^*. \tag{47}$$

That is, KKT conditions can be met by scheduling the user with

the largest $\{U'_m(r_m)+\mu_m+\phi_m-\pi\}r_{mn}+\lambda_m$ for slot n, and this gives a global maximum because this is a convex problem. □

B. Proof of Theorem 2

Proof: For any feasible scheduling policy Q satisfying the QoS constraints, we have

$$\sum_{m=1}^{M}\left[\left\{U'_m\left(\overline{R}_m^{Q^*}\right)+\mu_m^*+\phi_m^*-\pi\right\}R_m+\lambda_m^*\right]I_{\{Q=m\}} \le \sum_{m=1}^{M}\left[\left\{U'_m\left(\overline{R}_m^{Q^*}\right)+\mu_m^*+\phi_m^*-\pi\right\}R_m+\lambda_m^*\right]I_{\{Q^*=m\}} \tag{48}$$

where $\overline{R}_m^{Q^*}$ is user m's average throughput obtained by scheduler Q^*. Considering the expectations on both sides and manipulating them, we obtain

$$\begin{aligned}&\sum_{m=1}^{M}\left[\left\{U'_m\left(\overline{R}_m^{Q^*}\right)+\mu_m^*+\phi_m^*-\pi\right\}\overline{R}_m^{Q}+\lambda_m^*\Pr\{Q=m\}\right]\\ &\le\sum_{m=1}^{M}\left[\left\{U'_m\left(\overline{R}_m^{Q^*}\right)+\mu_m^*+\phi_m^*-\pi\right\}\overline{R}_m^{Q^*}+\lambda_m^*\Pr\{Q^*=m\}\right]\end{aligned} \tag{49}$$

$$\begin{aligned}\Leftrightarrow&\sum_{m=1}^{M}U'_m\left(\overline{R}_m^{Q^*}\right)\left(\overline{R}_m^{Q}-\overline{R}_m^{Q^*}\right)\\ &\le-\sum_{m=1}^{M}\lambda_m^*\Big[(\Pr\{Q=m\}-\alpha_m)-(\Pr\{Q^*=m\}-\alpha_m)\Big]\\ &\quad-\sum_{m=1}^{M}\mu_m^*\left[\left(\overline{R}_m^{Q}-\beta_m\right)-\left(\overline{R}_m^{Q^*}-\beta_m\right)\right]\\ &\quad-\sum_{m=1}^{M}\phi_m^*\left(\overline{R}_m^{Q^*}-\gamma_m\sum_{i=1}^{M}\overline{R}_m^{Q^*}\right)-\sum_{m=1}^{M}(\phi_m^*-\pi)\overline{R}_m^{Q}.\end{aligned} \tag{50}$$

$$\begin{aligned}\Leftrightarrow&\sum_{m=1}^{M}U'_m\left(\overline{R}_m^{Q^*}\right)\left(\overline{R}_m^{Q}-\overline{R}_m^{Q^*}\right)\\ &\le-\sum_{m=1}^{M}\lambda_m^*(\Pr\{Q=m\}-\alpha_m)-\sum_{m=1}^{M}\mu_m^*\left(\overline{R}_m^{Q}-\beta_m\right)\\ &\quad-\sum_{m=1}^{M}(\phi_m^*-\pi)\overline{R}_m^{Q}.\end{aligned} \tag{51}$$

In (49) and (50), we used the conditions as follows:

$$\begin{aligned}&\lambda_m^*(Pr\{Q^*=m\}-\alpha_m)=0,\\ &\mu_m^*\left(\overline{R}_m^{Q}-\beta_m\right)=0,\\ &\phi_m^*\left(\overline{R}_m^{Q}-\gamma_m\sum_{m=1}^{M}\overline{R}_m^{Q}\right)=0.\end{aligned}$$

Since Q satisfies the QoS constraints, the inequality becomes

$$\sum_{m=1}^{M}U'_m\left(\overline{R}_m^{Q^*}\right)\left(\overline{R}_m^{Q}-\overline{R}_m^{Q^*}\right)\le 0. \tag{52}$$

Using the vector notations, we get

$$\nabla U(\overline{\mathbf{R}}_{Q^*})(\overline{\mathbf{R}}_Q-\overline{\mathbf{R}}_{Q^*})\le 0 \tag{53}$$

where $\nabla U(\overline{\mathbf{R}})=(\mathbf{U}'_{\mathbf{1}}(\overline{\mathbf{R}}_{\mathbf{1}}),\cdots,\mathbf{U}'_{\mathbf{M}}(\overline{\mathbf{R}}_{\mathbf{M}}))$, $\overline{\mathbf{R}}_Q=(\overline{R}_1^Q,\cdots,\overline{R}_M^Q)$, and $\overline{\mathbf{R}}_{Q^*}=(\overline{R}_1^{Q^*},\cdots,\overline{R}_M^{Q^*})$. This means that $U(\overline{\mathbf{R}})$ has the maximum at $\overline{\mathbf{R}}=\overline{\mathbf{R}}_{\mathbf{Q}^*}$. □

C. Proof of Lemma 9

Proof: For updating λ, we use (26) which always produces non-negative values due to maximum function. Before proving the convergence of the algorithm, we consider a similar algorithm that does not apply maximum function.

$$\lambda_{m,k+1}=\lambda_{m,k}+\delta_k^{\lambda}\left[\alpha_m-I_{m,k}\right] \tag{54}$$

where $\delta_k^{\lambda}=1/k$. From the updating algorithm, we can define

$$\overline{g}_m^{\lambda}(\lambda)=E\left[\alpha_m-I_{\left\{F_m\left(\lambda;\overline{\mathbf{R}},\mu,\phi\right)\ge F_j\left(\lambda;\overline{\mathbf{R}},\mu,\phi\right),j\ne m\right\}}\right]. \tag{55}$$

If $\overline{g}_m^{\lambda}(\lambda)$ satisfies K-condition, λ converges weakly to the unique solution of the following.

$$\dot{\lambda}_m=\overline{g}_m^{\lambda}(\lambda). \tag{56}$$

There are two arbitrary vectors having the relation of $\hat{\lambda}\le\tilde{\lambda}$ and $\hat{\lambda}_m=\tilde{\lambda}_m$. For these vectors, function F satisfies

$$F_j\left(\hat{\lambda};\overline{\mathbf{R}},\mu,\phi\right)\begin{cases}\ge F_j\left(\tilde{\lambda};\overline{\mathbf{R}},\mu,\phi\right) & j\ne m,\\ =F_j\left(\tilde{\lambda};\overline{\mathbf{R}},\mu,\phi\right) & j=m.\end{cases} \tag{57}$$

Then, we obtain the following inequality.

$$\begin{aligned}\overline{g}_m^{\lambda}\left(\hat{\lambda}\right)&=E\left[\alpha_m-I_{\left\{F_m\left(\hat{\lambda};\overline{\mathbf{R}},\mu,\phi\right)\ge F_j\left(\hat{\lambda};\overline{\mathbf{R}},\mu,\phi\right),j\ne m\right\}}\right]\\ &\le E\left[\alpha_m-I_{\left\{F_m\left(\tilde{\lambda};\overline{\mathbf{R}},\mu,\phi\right)\ge F_j\left(\tilde{\lambda};\overline{\mathbf{R}},\mu,\phi\right),j\ne m\right\}}\right]\\ &=\overline{g}_m^{\lambda}\left(\tilde{\lambda}\right).\end{aligned} \tag{58}$$

So, $\overline{g}_m^{\lambda}(\lambda)$ satisfies K-condition, and the convergence of (54) is proved. If the algorithm converges to non-negative value, the original algorithm (26) converges to the non-negative value, though then convergence speeds of them may be different. In the opposite case, the original algorithm converges to zero, so its convergence is proved. □

REFERENCES

[1] F. P. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.

[2] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness, and stability," *J. Operational Research Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[3] A. Jalali, R. Padovani, and P. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE VTC*, vol. 3, May 2000, pp. 1854–1858.
[4] J. M. Holtzman, "Asymptotic analysis of proportional fair algorithm," in *Proc. IEEE PIMRC*, vol. 2, 2001, pp. F33–F37.
[5] S. Borst and P. Whiting, "Dynamic rate control algorithms for HDR throughput optimization," in *Proc. IEEE INFOCOM*, 2001, pp. 976–985.
[6] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting,"CDMA data QoS scheduling on the forward link with variable channel conditions," *Bell Lab. Tech. Rep.*, Apr. 2000.
[7] S. Shakkottai and A. L. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in HDR," in *Proc. International Teletraffic Congress*, Sept, 2000, pp. 793–804.
[8] M. Agarwal and A. Puri, "Base station scheduling of requests with fixed deadlines," in *Proc. IEEE INFOCOM*, 2002, pp. 487–496.
[9] D. Wu and R. Negi,"Downlink scheduling in a celluar network for quality-of-service assurance," *IEEE Trans. Veh. Technol.*, vol. 53, no. 5, pp. 1547–1557, Sept. 2004.
[10] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Compu. Netw. J.*, vol. 41, no. 4, pp. 451–474, Mar. 2003.
[11] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power-control scheme in wireless cellular systems," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 210–221, Apr. 2003.
[12] C. Li, X. Wang, and D. Reynolds, "Utility-based joint power and rate allocation for downlink CDMA with blind multiuser detection," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1163–1174, May 2005.
[13] V. K. N. Lau, "Analytical framework for multiuser uplink MIMO space-time scheduling design with convex utility functions," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1832–1843, Sept. 2004.
[14] V. K. N. Lau, "Optimal downlink space-time scheduling design with convex utility functions-multiple-antenna systems with orthogonal spatial multiplexing," *IEEE Trans. Veh. Technol.*, vol. 54, no. 4, pp. 1322–1333, July 2005.
[15] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1250–1259, July 2004.
[16] Z. Jiang, Y. Ge, and Y. G. Li, "Max-utility wireless resource management for best-effort traffic," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 100–111, Jan. 2005.
[17] C. Curescu and S. Nadjm-Tehrani, "Time-aware utility-based resource allocation in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 7, pp. 624–636, July 2005.
[18] R. Agrawal, A. Bedekar, R. J. La, and V. Subramanian, "Class and channel condition based weighted proportional fair scheduler," in *Proc. Teletraffic Engineering in the Internet Era, ITC-17*, 2001,pp. 553–565.
[19] R. Agrawal and V. Subramanian, "Optimality of certain channel-aware scheduling policies," in *Proc. AACCCC*, 2002, pp. 1532–1541.
[20] D. M. Andrews, L. Qian, and A. L. Stolyar, "Optimal utility-based throughput allocation subject to throughput constraints," in *Proc. IEEE INFOCOM*, 2005, pp. 2415–2424.
[21] A. L. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation," Operations Research. no. 53, pp. 12–25, 2005.
[22] P. Viswananth, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1277–1294, June. 2002.
[23] H. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications*, Springer, New York, 1997.
[24] T. E. Kolding, "QoS-aware proportional fair packet scheduling with required activity detection," in *Proc. IEEE VTC*, Sept. 2006.

Neung-Hyung Lee received B.S., M.S., and Ph.D. degrees in the School of Electrical Engineering & Computer Science, Seoul National University, in 2000, 2002, and 2007, respectively. He is a Senior Engineer in Samsung Electronics. His research interests include resource management and packet scheduling in wireless networks, and next generation wireless networks.

Jin-Ghoo Choi received B.S., M.S., and Ph.D. degrees in the school of Electrical Engineering & Computer Science, Seoul National University in 1998, 2000, and 2005, respectively. From 2006 to 2007, he worked for Samsung Electronics as a Senior Engineer. In 2009, he was with the Department of Electrical & Computer Engineering at Ohio State University as a Visiting Scholar. He joined the Department of Information and Communication Engineering in Yeungnam University as a Faculty Member in 2010. His research interests include performance analysis of communication networks, packet scheduling policy in wireless networks, and wireless sensor networks.

Saewoong Bahk received B.S. and M.B. degrees in Electrical Engineering from Seoul National University in 1984 and 1986, respectively, and a Ph.D. degree from the University of Pennsylvania in 1991. From 1991 through 1994, he was with the Department of Network Operations Systems at AT&T Bell Laboratories as an MTS where he worked for AT&T network management. In 1994, he joined the School of Electrical Engineering at Seoul National University and currently serves as a Professor. His interests include performance analysis of communication networks and network security.