Combined QoS Scheduling and Call Admission Control Algorithm in Cellular Networks

Hyang-Won Lee and Song Chong

***Abstract*— In this paper, a scheduling problem in wireless networks is considered when there simultaneously exist CBR (constant bit rate) users requiring exact minimum average throughput guarantee and EMG (elastic with minimum guarantee) users requiring minimum average throughput guarantee and more if possible. By exploiting utility maximization problem without minimum throughput constraint and newly defined utility functions, we propose a combined scheduling and call admission control scheme that exactly guarantees the minimum requirements of CBR and EMG users and then allocates the leftover capacity to EMG users. In the proposed scheme, it is easy to give priority to particular users so that they are guaranteed their requirements prior to any other user. Moreover, the priority structure enables the proposed measurement-based call admission control algorithm to perform admission trial without affecting the minimum performance of existing users. We show through mathematical analysis and simulations that our scheme works as designed.**

I. Introduction

Quality of service (QoS) in wireless networks has become a very important issue as the wireless systems supporting high data rates emerge and the type of supported applications gets diverse. For example, multimedia applications usually require maximum tolerable delay and error probability and minimum throughput guarantee to receive acceptable quality of the applications. For QoS guarantee in wireless networks, many scheduling algorithms have been proposed. In such QoS schedulers, it is important not only to guarantee the minimum requirements of users but also to cope with the case where not all the requirements can be satisfied. If the infeasible case happens, it is desirable that the scheduler guarantees the users in the order of predetermined priorities or new arrivals are blocked to maintain feasibility.

In [1] and [2], the authors present two scheduling algorithms including M-LWDF (modified largest weighted delay first) and EXP. At each time t, the M-LWDF algorithm selects the user having the maximum decision metric[1] $\gamma_i r_{i,t+1} W_i(t)$ where γ_i is a positive constant, $r_{i,t+1}$ is the achievable data rate of user i during the time interval $[t, t+1)$, and $W_i(t)$ is the head-of-line (HOL) delay of user i's queue. The scheduler obviously gives priority to the user with high HOL delay and good channel condition. By properly setting the value of γ_i, the scheduler is shown to probabilistically guarantee each user's maximum tolerable delay. The EXP algorithm has a more complex form of decision metric and is shown to yield better delay performance than the M-LWDF algorithm [1]. They also show through simulations that if the two schedulers are combined with token counter, they can provide minimum throughput guarantee.

By appealing to stochastic optimization, Liu et al. propose a general scheduling framework in which the sum of each user's expected utility is maximized under several types of fairness constraints or minimum performance constraints [3]. Since the fairness constraints in the paper are always feasible, there always exists a solution to the problem with those constraints. Furthermore, it is shown for minimum time-fraction constraints that the average performance of each user at the solution is no worse than that of any non-opportunistic scheduling scheme. However, as the authors mention in the paper, the minimum performance constraints, which lead to QoS guarantee, incur the feasibility problem which is not easy to deal with.

In [4], Andrews et al. consider a concave utility maximization problem with minimum and maximum rate($R_i^{\min}$ and $R_i^{\max}$) constraints. In spite of the constraints which are hard to know the feasibility, they propose a solution to the problem, i.e., scheduling algorithm by modifying the token counter suggested in [1]. In the paper, two specific forms of the scheduling algorithm are shown to guarantee $R_i^{\min}$ and $R_i^{\max}$.

In this paper, we consider a wireless system where there exist two classes of users or applications. One is CBR (constant bit rate) applications which generate their data at a fixed rate, e.g., voice. The performance of such applications severely degrades if the minimum throughput (usually encoding rate) is not guaranteed, so they are likely to require minimum throughput guarantee. But, allocating more throughput than the requirement is nothing but waste of resource because the source transmission rate is always fixed at the requirement. The other is EMG (elastic with minimum guarantee) applications which require minimum throughput guarantee and require more if possible. For example, MPEG-4 FGS (fine granularity scability) enables to freely adjust the video rate to an arbitrary value in real time without time-consuming decoding and re-encoding operations, as long as the target rate is greater than or equal to that of the base layer [5]. Consequently, such applications would require minimum throughput guarantee for minimum acceptable performance and require more to enhance the performance if possible. Of course, the

This work was supported in part by the center for Broadband OFDM Mobile Access (BrOMA) at POSTECH through the ITRC program of the Korean MIC, supervised by IITA(IITA-2005-C1090-0502-0008) and in part by LG Electronics Inc..

The authors are with the Department of Electrical Engineering and Computer Science at Korea Advanced Institute of Science and Technology (e-mail: mslhw@netsys.kaist.ac.kr; song@ee.kaist.ac.kr).

[1]The value that a scheduler compares to decide which user to serve.

0-7803-9550-6/06/$20.00 © 2006 IEEE

data traffic of premium user can require such QoS guarantee. Note that the elastic traffic, which is one of the important application types [6], can be categorized into EMG class with zero minimum requirement. The users in CBR and EMG class excluding elastic users will be called QoS users throughout the paper.

From the viewpoint of the system of our interest, the above previous works have several weaknesses. Firstly, they do not describe how to deal with the capacity which remains after guaranteeing the requirements of QoS users. Although it is shown in [1] that the leftover capacity is allocated to non-real time users, the authors do not completely describe how the allocation is accomplished. Secondly, it seems to be difficult for them to handle the case where the capacity is not enough and thus not all the requirements can be satisfied. Note that even if they adopt call admission control (CAC) to maintain feasibility, the feasibility can be broken due to the time-varying capacity. In this case, if each user has a predetermined priority, the capacity will be distributed according to each user's priority so that the requirements of the users with high priorities will be fulfilled prior to any other user. But, to do this, two conditions should be satisfied. One is that the decision metrics of the users with high priorities should be greater than those of other users before the requirements of the high-priority users are fulfilled. The other is that if the requirement of a QoS user is fulfilled, the decision metric of the user should drop to zero so that the remaining capacity can be distributed to other QoS users of which the requirements are not satisfied yet. However, it is difficult to meet the two conditions with the previous scheduling schemes. Thirdly, QoS scheduling problem has been widely studied in the previous works and in the literature, but CAC problem has been hardly discussed. Since CAC is necessary for QoS guarantee, it is important to develop CAC algorithm operating with QoS scheduling algorithm.

By exploiting utility maximization problem without minimum throughput constraint and newly defined utility functions, we propose a combined scheduling and CAC scheme that guarantees the minimum requirements of QoS users and then allocates the leftover capacity to the users in EMG. In the proposed scheme, it is easy to give *priority* to particular QoS users so that they are guaranteed their requirements prior to any other user when not all the requirements can be satisfied. Therefore, if the infeasible case happens, our scheduler fulfills the requirements of the users according to their priorities. This feature is important because even if we adopt CAC to maintain feasibility, it can be broken due to time-varying wireless channels. Moreover, our scheduler achieves *PF^z-like share of leftover capacity* in the sense that the users with extremely bad channels achieve zero throughput and other users share the leftover capacity in such a way that a user with better channel condition yields higher throughput but with some degree of fairness. Our CAC algorithm is carried out based on measurement, i.e., it accepts and serves a new arrival and decide to admit or block the arrival after certain trial period. In such a measurement-based call admission control, it is very important that the admission trial does not deteriorate the performance of existing users. By taking advantage of the priority structure, the proposed *measurement-based call admission control algorithm* can perform the admission trial without affecting the minimum performance of existing QoS users.

The rest of the paper is structured as follows. In Section II, we discuss the motivation of this work and the background needed to understand this paper. In Section III, we propose a scheduling and CAC algorithm that achieves the objective, and mathematically analyze the properties of the algorithm. Our algorithm is examined in Section IV, and the limitations and extensions of our work are discussed in Section V. Finally, we conclude the paper in Section VI.

II. Motivation and Background

A. Utility Maximization Problem

Since the pioneering work [7] which adopted utility maximization problem for network flow control was published, the utility maximization framework has been applied to many areas both in wired networks and in wireless networks. Recently, Kushner et al. analyzed the optimality and convergence of PF scheduler [8] based on the utility maximization problem and some standard results in stochastic approximation [9]. We summarize the result of the paper here.

Consider a time-slotted wireless system, and let $R_i(t)$ be the average throughput of user i up to time t. Then, $R_i(t)$ is given by

$$R_i(t) = \frac{\sum_{\tau=1}^{t} r_{i,\tau} I_{i,\tau}}{t} \tag{1}$$

where $r_{i,\tau+1}$ is the achievable data rate of user i during $[\tau, \tau+1)$, i.e., $(\tau+1)$-th time slot, and $I_{i,\tau+1}$ is the indicator function such that $I_{i,\tau+1} = 1$ if user i is chosen at time τ to be served in slot $\tau+1$ and $I_{i,\tau+1} = 0$ otherwise. (1) can be rewritten in iteration form as

$$R_i(t+1) = R_i(t) + \epsilon_t \left[r_{i,t+1} I_{i,t+1} - R_i(t) \right] \tag{2}$$

where $\epsilon_t = \frac{1}{t+1}$. Define $U(R(t)) = \sum_i \log(R_i(t))$, then by first order Taylor expansion in the neighborhood of $\epsilon_t = 0$, we have

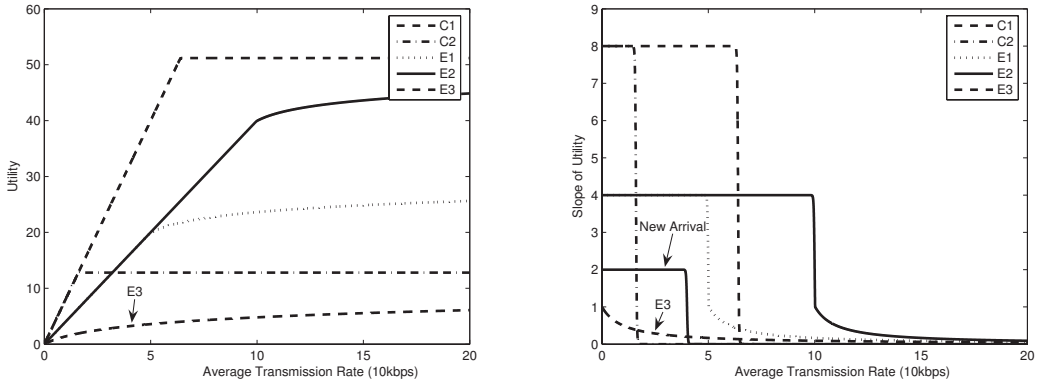
$$U(R(t+1)) - U(R(t)) = \epsilon_t \sum_i \frac{r_{i,t+1} I_{i,t+1} - R_i(t)}{R_i(t)} + O(\epsilon_t^2), \tag{3}$$

of which the derivation is shown for general utility functions in Appendix II. Obviously, selecting user i^* such that

$$i^* = \arg\max_i \frac{r_{i,t+1}}{R_i(t)} \tag{4}$$

maximizes the first order term and as $\epsilon_t \to 0$, the scheduler results in maximizing $\lim_{t\to\infty} U(R(t+1)) - U(R(t))$.

For the proof of convergence, they first show that the limit point of the iteration (2) corresponding to (4) weakly converges to the set of limit points of the solution of an ODE (ordinary differential equation). After that, the existence, uniqueness, and global asymptotic stability of the limit points



(a) Utility functions

(b) Slope of utility functions

Fig. 1. Utility function and its derivative

of the ODE are proved. For the optimality, they use the strict concavity of logarithmic utility function and show that the scheduler (4) maximizes $\lim_{t\to\infty} U(R(t))$. As well-known, (4) is nothing but PF scheduler and therefore PF scheduler achieves proportional fairness. All the above results hold even if the average computation (1) is replaced by EWMA (exponentially weighted moving average). Moreover, all the above results can be extended to the algorithms based on any strictly concave utility function [10].

B. Minimum Guarantee

For minimum average throughput guarantee, one might want to simply add minimum constraints to the utility maximization problem. Obviously, the solution to the problem guarantees the minimum throughput if exists, but it is very difficult not only to find the solution in time-slotted systems but also to investigate the feasibility of the problem. In practice, CAC algorithm is adopted to maintain feasibility, and a scheduling algorithm is used to provide the minimum throughput guarantee. In other words, CAC is necessary in order for QoS scheduler to work as desired. Then, we may ask two questions: one is "Is it possible to guarantee minimum average throughput by simply solving utility maximization problem without minimum throughput constraint?" and the other is "Is it possible to jointly perform CAC and QoS scheduling?". Our answer to these two questions is "Yes", and we will propose a combined scheduling and CAC algorithm based on utility maximization problem without minimum throughput constraint and new utility functions defined first in this paper.

III. Proposed Algorithm

Traditionally, the elastic traffic such as FTP is modeled as a strictly concave utility function and the hard real-time traffic(or minimum-guarantee application) is modeled as a step utility function [11], which is usually approximated by a sigmoidal function for mathematical tractability. Since the analysis by Kushner holds true only with strictly concave utility functions, we cannot use the conventional utility functions for our objective.

A. New Utility Function

Instead of using the conventional utility functions as they are, we would like to redefine the utility functions of QoS users as strictly concave ones. Let $S = C \cup E$ where C and E are the set of CBR users and EMG users, respectively. Let R_i be the average throughput of user i, then we define the utility functions as: for $i \in C$,

$$U_i(R_i) = c_i \left\{ 1 - \frac{\log\left(1 + e^{-b_i(R_i - m_i)}\right)}{\log\left(1 + e^{b_i m_i}\right)} \right\} \tag{5}$$

and for $i \in E$,

$$U_i(R_i) = \begin{cases} c_i \left\{ 1 - \frac{\log\left(1+e^{-b_i(R_i-m_i)}\right)}{\log\left(1+e^{b_i m_i}\right)} \right\}, & R_i < m_i^\delta \\ a_i \log\left(1 + R_i - m_i^\delta\right) + \Delta_i, & R_i \geq m_i^\delta \end{cases} \tag{6}$$

where a_i, b_i, c_i and Δ_i are positive constants, m_i is the minimum demand rate of user i, and $m_i^\delta = m_i + \delta_i$ where δ_i is a small nonnegative constant. a_i and c_i are determined according to user i's priority. Especially, a_i is set to be equal for all $i \in E$ and we will explain the detail below. b_i is set to be equal for all $i \in S$ and see Appendix I-C for the role of b_i. With these a_i, b_i and c_i, the values of Δ_i and δ_i are determined such that the continuity of utility function and its derivative holds. See Appendix I-A for the detail and I-B for the strict concavity of the new utility functions. Notice that if $m_i^\delta = 0$ in (6), the utility function corresponds to elastic users, so (6) is a generalized version of the conventional utility function for elastic traffic. We will let $m_i^\delta = 0$ for elastic user.

For simplicity of exposition, the QoS class and its utility function are specified together by $C_i(m_i, c_i, b_i)$ or $E_i(m_i, c_i, b_i, a_i)$. For example, $C_1(10, 5, 50)$ stands for CBR class 1 of which the utility function is given by (5) with parameters $m_1 = 10$, $c_1 = 5$ and $b_1 = 50$. Fig. 1(a) shows an example of the new utility functions corresponding to: $C_1(6.4, 51.2, 50)$, $C_2(1.6, 12.8, 50)$, $E_1(5, 20, 50, 1)$, $E_2(10, 40, 50, 1)$ and $E_3(0, 0, 0, 1)$. Note that the users in E_3 are elastic users. As seen in the figure, for the users in CBR class, the utility does not increase above the minimum

requirement. Thus, if the utility function is used in utility maximization problem, the allocated throughput will not be higher than the requirement. On the other hand, the utility function of the users in EMG class increases above the minimum requirements. Accordingly, the throughput allocated to EMG users by utility maximization can be higher than their requirements if there is surplus capacity after satisfying all the minimum requirements. Following to this property, we call the region $R_i \geq m_i^\delta$ for EMG users *elastic part* and the region $R_i < m_i^\delta$ *inelastic part*. Note that elastic user has only elastic part and unless otherwise specified, the terminology "elastic part" or "elastic part of EMG user" contains elastic user.

The slopes of the functions are shown in Fig. 1(b) from which we can see that the slopes of CBR users are higher than those of EMG users below the requirements of CBR users. Obviously, this gives higher priority to CBR users in maximizing the sum of utility functions because allocating the throughput to EMG users results in less increase of the objective function than allocating to CBR users. Therefore, we expect the minimum requirement of CBR users will be satisfied by the utility maximization problem prior to EMG users. Moreover, the slope sharply drops down to zero above the minimum requirement so that the remaining capacity will be allocated to EMG users. For the remaining capacity, the minimum requirements of the users in E_1 and E_2 will be similarly fulfilled, and if there remains the capacity after that, E_1, E_2 and E_3 will share the leftover capacity.

B. Scheduling Algorithm and Its Analysis

For given utility functions, we can easily derive the scheduling policy that maximizes the sum of the utility functions as shown in Appendix II. The decision metric for general utility functions is given by $r_{j,t+1}U_j'(R_j(t)), \forall j \in S$, and using the metric, the scheduler will select user j^* at time t such that

$$j^* = \arg\max_j r_{j,t+1}U_j'(R_j(t)) \tag{7}$$

where $U_j'(R_j(t))$ is given as

$$U_j'(R_j(t)) = \frac{b_i c_i}{\log\left(1+e^{b_i m_i}\right)} \cdot \frac{e^{-b_i(R_j(t)-m_i)}}{1+e^{-b_i(R_j(t)-m_i)}} \tag{8}$$

for $j \in C_i(m_i, c_i, b_i)$ and

$$U_j'(R_j(t)) = \begin{cases} \frac{b_i c_i}{\log\left(1+e^{b_i m_i}\right)} \cdot \frac{e^{-b_i(R_j(t)-m_i)}}{1+e^{-b_i(R_j(t)-m_i)}}, & R_j(t) < m_i^\delta \\ \frac{a_i}{1+R_j(t)-m_i^\delta}, & R_j(t) \geq m_i^\delta. \end{cases} \tag{9}$$

for $j \in E_i(m_i, c_i, b_i, a_i)$. The optimality and convergence of the scheduler can be readily proved following the results in [9] because the utility functions are strictly concave.

In this paper, we analyze the properties of the limit point of algorithm (7) with respect to our objectives mentioned in Section I. Let $R_i = \lim_{t\to\infty} R_i(t), \forall i$, and β_i be the mean rate of user i. Then, $\frac{R_i}{\beta_i}$ is the fraction of times during which user i has been served, and the sum of such fractions for all users should not exceed 1, i.e., $\sum_i \frac{R_i}{\beta_i} \leq 1$. As a capacity region, we use the intersection of the time fraction constraint and $R \geq 0$ for analytical tractability. Note that the capacity region we will use is actually the convex hull of $R = 0$ and corner points $[R_i = \beta_i$ and $R_j = 0, \forall j \neq i], \forall i$, and thus it is the approximation of the real capacity region.

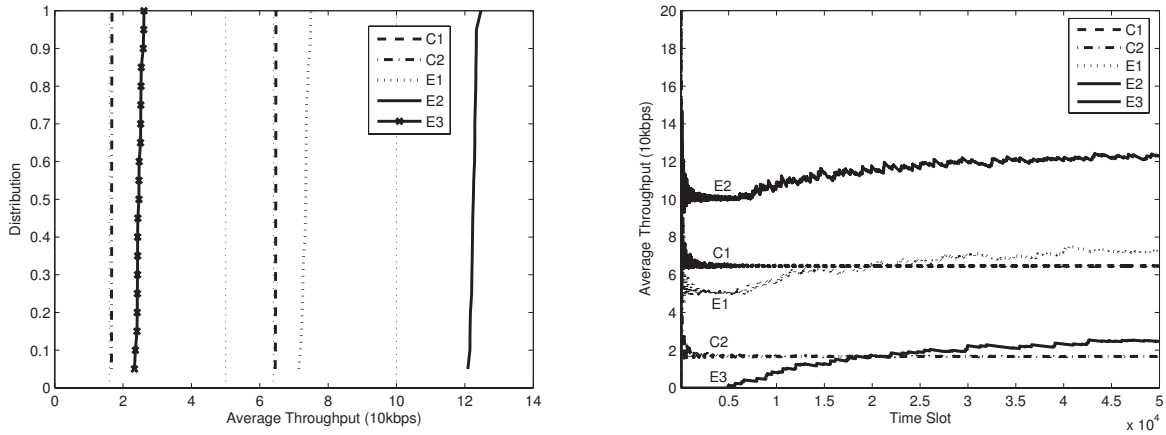
Theorem 3.1: Assume that the feasible region of $R_i, \forall i$ is represented as $\sum_i \frac{R_i}{\beta_i} \leq 1$ and $R \geq 0$. Then, if $\beta_i U_i'(R_i) \geq \beta_j U_j'(R_j)$ for $R_i \leq m_i$, the minimum requirement of user i will be guaranteed when only one of m_i and m_j can be satisfied.

See Appendix III for the proof. For homogeneous channels ($\beta_i = \beta_j$), the above theorem shows that we can give priority to user i by setting the derivative as $U_i'(R_i) \geq U_j'(R_j)$ for $R_i \leq m_i$. Thus, the utility functions in Fig. 1 set the priority relationship as $C_1, C_2 > E_1, E_2 > E_3$, and this is what we expected in the above section. For the case of heterogeneous channels, if the height of $U_i'(R_i)$ is set to a sufficiently large value, then the inequality can be satisfied so that the priority relationship still holds. As discussed in Appendix I-C, the height of the slope of our new utility function can be arbitrarily set by only adjusting the value of c_i without changing the drop-down property. Thus, the priority relationship between users can be arbitrarily established irrespective of channel conditions, even for the users requiring the same minimum average throughput. This property is very important because even if we adopt CAC to maintain feasibility of users' QoS requirements, the feasibility can be broken due to the time-varying capacity of wireless channel. Our scheduler can cope with this infeasibility problem by guaranteeing the requirements in the order of predetermined priorities, and this is quite reasonable because the user with higher priority will pay more money. In Section IV, we verify this argument through simulation results.

On the share of leftover capacity, we can prove the following theorem of which the proof is shown in Appendix IV.

Theorem 3.2: Let $f_j(R_j)$ be the elastic part of EMG user j, i.e., $f_j(R_j) = U_j(R_j)$ for $R_j \geq m_j^\delta$, and suppose that for any QoS user i and any EMG user j, it holds $\beta_i U_i'(R_i) \geq \beta_j f_j'(R_j)$ for $R_i \leq m_i$. Then, all the minimum requirements of QoS users are fulfilled and after that, the leftover capacity is shared by EMG users in PF^z-like manner.

Theorem 3.2 implies that any elastic part of EMG user cannot achieve nonzero throughput unless all the minimum requirements of QoS users are fulfilled. Moreover, if the capacity remains after fulfilling all the requirements, EMG users will share the leftover capacity in PF^z-like manner, i.e., an elastic part with better channel condition will achieve higher throughput with some degree of fairness, but an elastic part with extremely bad channel condition will get no throughput. We can make $\beta_j U_j'(R_j)$ equal for every elastic user j by adjusting a_j so that they have the same priority. But, it is hard to do that because β_i is difficult to know in practice. Note however that for the priority of QoS users, we only need to set c_i to a sufficiently large value irrespective of β_i, which is easy to do.



(a) Distribution of average throughput

(b) Average throughput over time

Fig. 2. Simulation results with $W = 1$MHz: minimum guarantee and convergence

C. Call Admission Control Algorithm

We propose a measurement-based call admission control algorithm combined with the scheduling algorithm shown above. The following theorem shows that with our scheduling algorithm, it is possible to develop a CAC algorithm which performs admission trial without affecting the minimum performance of existing QoS users.

Theorem 3.3: Suppose that user k arrives at the system. If $U_k(R_k)$ is selected such that $\beta_k U_k'(R_k) \leq \beta_i U_i'(R_i)$ for $R_i \leq m_i$ for every existing QoS user i, then serving user k according to (7) does not violate the minimum guarantee of existing QoS users.

Proof: The proof is quite straightforward following Theorem 3.1. ■

Based on the above theorem, we suggest a CAC algorithm as follows. When a new call arrives, it is admitted and served by using predefined utility function for admission trial. As an example for homogeneous channels, we show the derivative of the utility function when the minimum requirement is 4 in Fig. 1(b). As seen in the figure, the derivatives of existing QoS users is higher than that of the new arrival below their minimum requirements so that serving the new arrival does not deteriorate the minimum performance of existing QoS users. On the other hand, the throughput of elastic parts of EMG users will probably decrease by the admission trial, which is not unacceptable according to the properties of EMG users. If the average throughput of the new arrival after certain trial period satisfies its minimum requirement, then the new arrival is admitted. Otherwise, it is blocked.

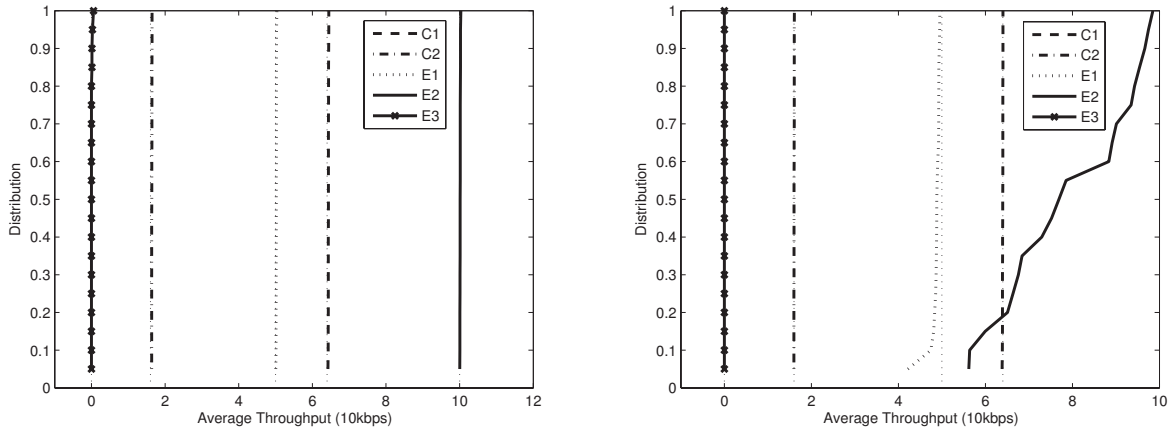
Once it is decided to be admitted, the parameters of the new arrival are changed to originally intended values. For example, if a new user belonging to $C_i(6.4, 51.2, 50)$ arrives, it will be firstly served with the parameters $m_i = 6.4$, $c_i = 12.8$ and $b_i = 50$. Right after it is admitted, the value of c_i is changed to 51.2, which corresponds to the originally intended priority. By giving the lowest priority to the new arrivals under admission trial in this way, we can assure that the admission trial does not affect the minimum performance of existing QoS users. For another example, if a new user in $E_1(5, 20, 50, 1)$ arrives, it will be firstly served by the scheduler (7) and (8) with the parameters $m_i = 5$, $c_i = 10$ and $b_i = 50$. When it is decided to be admitted, the value of c_i is changed to 20 and the user will be served by the scheduler (7) and (9). Note that the purpose of CAC is to test the feasibility of minimum requirements and thus EMG users are served like CBR users during admission trial period.

IV. Simulation Results

In this section, we demonstrate through simulations that our algorithm works as designed. The results are broken into two parts including, the case of homogeneous channels and the case of heterogeneous channels, and we will examine the characteristics of minimum guarantee, priority, share of leftover capacity and CAC. There are 5 classes including $C_1(64kbps, 51.2, 50)$, $C_2(16kbps, 12.8, 50)$, $E_1(50kbps, 20, 50, 1)$ and $E_2(100kbps, 40, 50, 1)$ and $E_3(0, 0, 0, 1)$, and each class has 20 users. The utility functions and their derivatives are equivalent to those shown in Fig. 1. We assume that the achievable data rate $r_{i,\tau}$ is given as Shannon bound, i.e., $r_{i,\tau} = W \log_2(1 + S_i/N_i)$ where W, S_i and N_i are the bandwidth of the channel, received signal power and total noise power, respectively.

A. Homogeneous Channel

Under homogeneous Rayleigh fading channels, we examine the performance of our scheduler by varying W in the Shannon bound. For the generation of the channels, we used MATLAB simulation files [12], and the simulation was run over 50000 time slots. Fig. 2 shows the distribution of the average throughput allocated to all users and the average throughput over time when $W = 1$MHz. As seen in Fig. 2(a), CBR users are exactly guaranteed their minimum average throughput requirements while as EMG users are guaranteed *minimum plus some share of leftover capacity*. Observe that



(a) Distribution of average throughput with $W = 0.8$MHz

(b) Distribution of average throughput with $W = 0.6$MHz

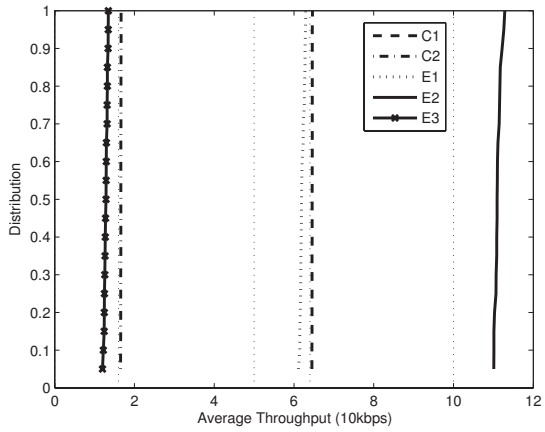
Fig. 3. Simulation results with different W's: priority

for EMG users, the values $R_i(t) - m_i$ are almost equal and this is the desired result because the PF^z-like share is equal for all users under homogeneous channels. We present the average throughput of five users (one user from each class) over time in Fig. 2(b), from which we can see that those of CBR users converge to their minimum requirements and those of EMG users converge to minimum + PF^z-like share although the convergence speed is low.

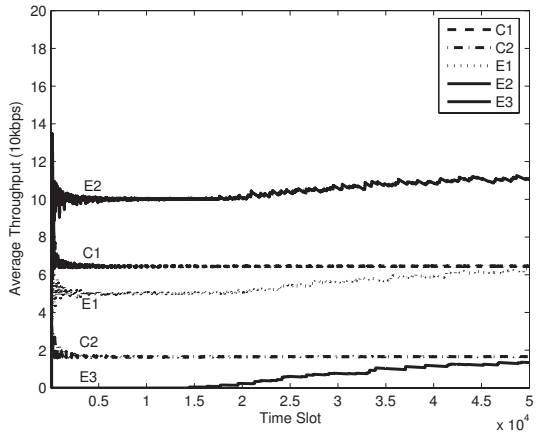
To verify the priority structure mentioned in the above section, we decrease the value of W and show the results with different W's in Fig. 3. Note that in practice, W does not change, but we change the value just to see what happens when the system capacity decreases. With $W = 0.8$MHz, the minimum requirements are exactly fulfilled for the users belonging to C_1, C_2, E_1 and E_2, and the elastic users (E_3) are allocated zero throughput as shown in Fig. 3(a). Comparing with Fig. 2(a), we can see that the performance of CBR has not changed, but the leftover capacity allocated to EMG users has decreased to zero. Fig. 3(b) illustrates the results when $W = 0.6$MHz. In this case, only the minimum requirements of CBR users are satisfied, and EMG users are not guaranteed their requirements. So the priority is given as $C_1, C_2 > E_1, E_2 > E_3$, and this is exactly what we expected in the above section.

We test our combined scheduling and call admission control algorithm under the scenario where the static users exist as above and new calls or users arrive at the system. The admission trial period is set to 3000 slots, and a new call arrives 100 slots after the decision on the previous arrival is made. The first new call arrives at 10000-th time slot. New users arrive in the order of C_1, C_2, E_1 and E_2, i.e., the sequence of the minimum requirements of new arrivals is 64kbps(CBR), 16kbps(CBR), 50kbps(EMG), 100kbps(EMG), 64kbps(CBR), and so on. In order for the new users under admission trial to be at the lowest priority level, the height of $U_i'(R_i)$ of every new user i under trial is set to 2, i.e., $c_i = 2m_i$, which is actually half of the height of the lowest existing QoS class. See Fig. 1(b) for an example of such $U_i'(R_i)$. We assume that no other new calls arrive during admission trial, which is not impractical because the system can delay the admission trial on new calls for correct decision. W is set to 0.9MHz and Fig. 4(a) and 4(b) illustrate the results without new arrivals.

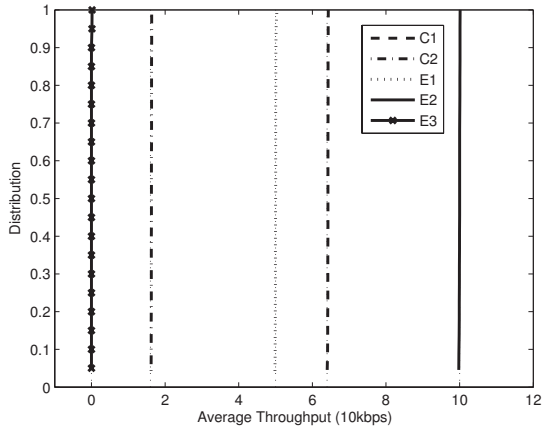
When there are new arrivals, the results are achieved as Fig. 4(c) and 4(d). First, Fig. 4(c) depicts the distribution of the average throughput of existing users and admitted arrivals. As seen in the figure, the minimum requirements are satisfied, which implies that our CAC maintains the feasibility of users' minimum requirements. The average throughput over time in Fig. 4(d) shows that the first 8 arrivals are admitted, the 9-th and 12-th arrivals are blocked, and the 10-th and 11-th arrivals are admitted. There can be two reasons for the block of the two arrivals. One is insufficient capacity and the other is insufficient time for the convergence of throughput to m_i. We simulated the scenario where the first to ninth users are static, and could see that all the minimum requirements are satisfied. Consequently, the reason for the block of the ninth arrival is the latter. In addition, simulating the scenario where the first to twelfth users are static shows that the feasibility is broken. Hence, our CAC algorithm works well in that it admits approximately the maximum number of feasible users. We remark that if the admission trial period is set to a larger value, our CAC will make more correct decision, but it will incur larger delay for admission trial. So, there is a tradeoff between the correctness of decision and the latency for decision, and consequently, the admission trial period can be selected according to system requirements. Lastly, observe from Fig. 4(d) that the minimum performance of existing QoS users is not affected by new arrivals, which is very important property for measurement-based CAC. In conclusion, the proposed combined scheduling and call admission control scheme guarantees the minimum average throughput requirements of QoS users, distributes the leftover capacity to EMG users, and performs admission control without deteriorating the minimum



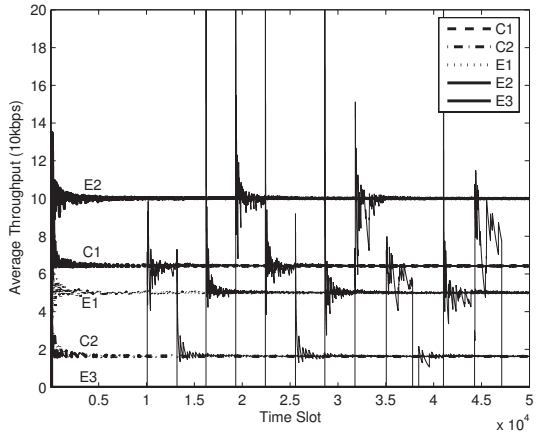
(a) Distribution of average throughput without new arrivals



(b) Average throughput over time without new arrivals



(c) Distribution of average throughput with new arrivals



(d) Average throughput over time with new arrivals

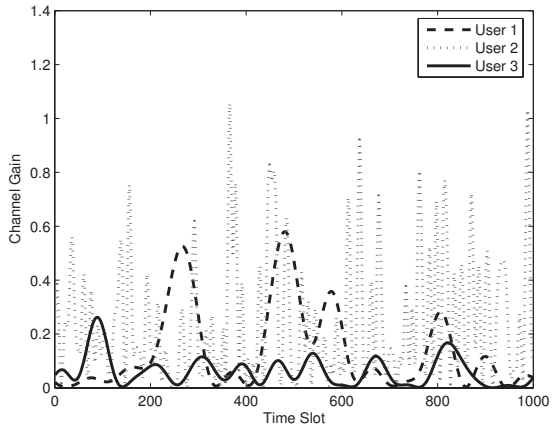
Fig. 4. Simulation results with $W = 0.9$MHz: call admission control

performance of existing QoS users.

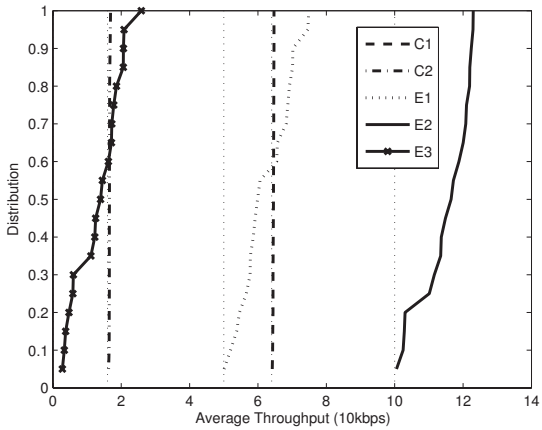
B. Heterogeneous Channel

The heterogeneous channels are generated by modifying the MATLAB simulation files used in the above subsection, and we present a sample of channel gains in Fig. 5(a). As seen in the figure, user 2 is in fast fading with high variance and high average gain while user 3 is in slow fading with low variance and low average. The distribution of throughput with $W = 2$MHz is shown in Fig. 5(b), from which we can see that CBR users are exactly guaranteed their minimum requirements and EMG users are also guaranteed their requirements. In contrast to the case of homogeneous channels, elastic parts of EMG users achieve different share from leftover capacity. For example, some EMG users are exactly guaranteed their minimum, i.e., their elastic parts yield zero throughput from leftover capacity while as other users yield nonzero throughput up to about 20kbps. This result is what we expected in the above section, and thus verifies our analysis.

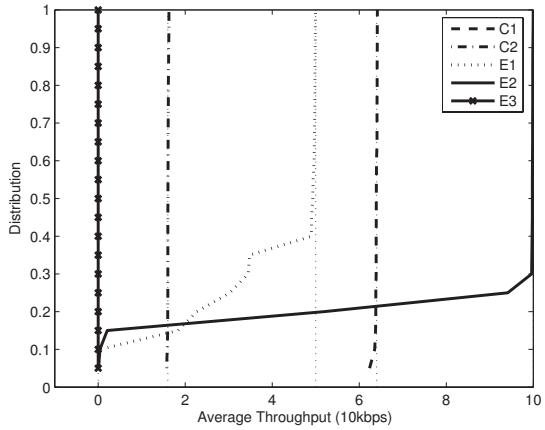
To examine the priority relationship, we decrease the value of W to 1MHz and show the distribution of throughput in Fig. 5(c). As seen in the figure, some users in C_1 are not guarantee their minimum, but some users in E_1 and E_2 are guarantee their minimum. This is undesirable because C_1 has higher priority than E_1 and E_2. According to our analysis in Section III, we can resolve this problem by setting the value of c_i to a sufficiently large value. To validate the analysis, we set the values of c_i's of C_1 and C_2 to 204.8 and 51.2 respectively, and show the distribution of throughput in Fig. 5(d). Observe that as desired, all the requirements of C_1 and C_2 having higher priority are satisfied, but not all the requirements of E_1 and E_2 are satisfied. Comparing with Fig. 5(c), we can see that the capacity allocated to some users in E_1 and E_2 has been reallocated to the users in C_1 who were not guaranteed before. In order to give higher priority to E_1, we set the value of c_i of E_1 to 160 and could see that all the requirements of C_1, C_2 and E_1 are satisfied although we do not show the result here due to limited space. This result shows that for any arbitrary channel, we can set the priority relationship by only adjusting the value of c_i.



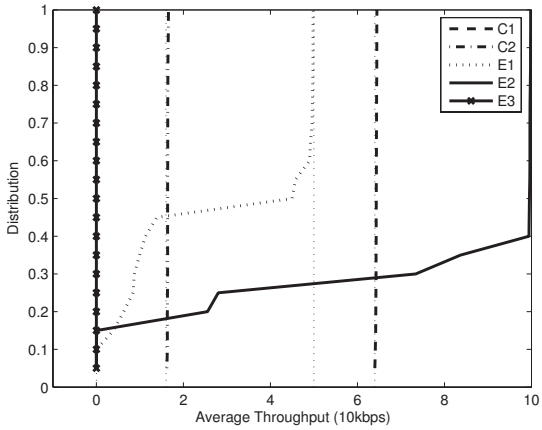
(a) Sample of channel gains



(b) Distribution of average throughput with $W = 2$MHz



(c) Distribution of Average throughput with $W = 1$MHz



(d) Distribution of average throughput with $W = 1$MHz and new c_i's

Fig. 5. Simulation results under heterogeneous channels

V. Limitations and Extensions

This paper is limited in a number of aspects:

- Applications usually have their peak rate constraints, but in our scheme, the constraints cannot be accommodated. This is problematic if a user is allocated higher throughput than its peak rate because it leads to the waste of resource.
- We assume that the traffic is persistent, i.e., every user always has the data to send. But in practice, some users can sometimes have empty queues even if they require minimum performance guarantee.
- If the utility function of elastic users is replaced by $\log(R_i)$ and the elastic part of EMG users is matched to the function, it is obvious that leftover capacity is shared in proportionally fair manner. However, if $\log(R_i)$ is used, elastic users will always achieve positive throughput because the derivative of the function is ∞ at $R_i = 0$. In order to open up the possibility that elastic users are allocated zero throughput when there is not enough capacity, we adopt $\log(1 + R_i)$. But, the fairness on the share of leftover capacity is hard to analyze, and we just call it PF^z-like share.
- For analytical tractability, we assume that the capacity region of R is approximately represented as a polyhedron. But precisely, the capacity region might be depicted as nonlinear shape, which is almost impossible to express in closed form.
- Although we proposed a call admission control algorithm, its optimality was not proved mathematically. If it can be proved that the admission result by the proposed CAC algorithm achieves the maximum number of feasible users, it would greatly benefit our work.

We can extend the results of this paper in several ways:

- As mentioned in Section I, multimedia applications usually require QoS guarantee. Those applications would demand not only minimum average throughput guarantee but also strict packet delay and delay jitter guarantee, but the proposed scheduler guarantees only the minimum average throughput. So, it would be more meaningful and practical work if we can extend the results of this

paper to jointly guarantee delay, delay jitter and average throughput.

- Recently, OFDM (Orthogonal Frequency Division Multiplexing) system has been adopted by most of the next generation wireless interface standard, which defines QoS guarantee. According to the trend, QoS guarantee in OFDM systems becomes an interesting issue and many researchers have begun to study the problem. We guess that our results can be easily extended to OFDM systems.
- Although this work deals with QoS guarantee in wireless networks, we can readily apply the results to QoS problem in wired networks.

VI. Conclusions

In this paper, we proposed a combined scheduling and call admission control scheme that guarantees the minimum average throughput requirements according to users' priorities, allocates the leftover capacity to EMG users in PF^z-like manner, and performs admission trial without deteriorating the minimum performance of existing users. We verified the performance of the proposed scheme through not only mathematical analysis but also simulation results. Moreover, the proposed algorithm performs as designed in heterogeneous channels as well as homogeneous channels.

Appendix I
Properties of New Utility Function

A. Continuity

For the continuity of $U_i'(R_i)$ in (9), the upper one and the lower one in (9) should be equal at $R_i = m_i^\delta$. By straightforward calculation, we can see that the continuity of $U_i'(R_i)$ holds when δ_i satisfies $a_i = \frac{b_i c_i}{\log\left(1+e^{b_i m_i}\right)} \frac{e^{-b_i\delta_i}}{1+e^{-b_i\delta_i}}$. By using δ_i computed from the equation, Δ_i is determined such that the continuity of $U_i(R_i)$ in (6) is satisfied, and it can be easily shown that $\Delta_i = c_i \left\{1 - \frac{\log\left(1+e^{-b_i\delta_i}\right)}{\log\left(1+e^{b_i m_i}\right)}\right\}$ leads to the continuity. Thus, $U_i(R_i)$ in (6) is a continuously differentiable function.

B. Strict Concavity of $U_i(R_i)$

The second derivative of $U_i(R_i)$ in (5) is given by

$$U_i''(R_i) = -\frac{b_i^2 c_i}{\log\left(1+e^{b_i m_i}\right)} \cdot \frac{e^{-b_i(R_i-m_i)}}{\left(1+e^{-b_i(R_i-m_i)}\right)^2} < 0$$

which implies that $U_i(R_i)$ in (5) is strictly concave. The strict concavity for $U_i(R_i)$ in (6) can also be proved easily. Therefore, $U_i(R_i)$ is strictly concave.

C. Shape of $U_i'(R_i)$ according to b_i and c_i

The derivative $U_i'(R_i)$ in C is given as (8), and its plot for $m_i = 4$ and different values of b_i and c_i is shown in Fig. 6. As seen in the figure, the slope with $b_i = 50$ drops to zero above m_i much more sharply than that with $b_i = 1$. Precisely, for large b_i, we have $U_i'(0) \approx \frac{c_i}{m_i}$, $U_i'(m_i - \xi) \approx \frac{c_i}{m_i}$, $U_i'(m_i) \approx \frac{c_i}{2m_i}$ and $U_i'(m_i + \xi) \approx 0$ where ξ is a very small positive value. Thus, R_i will not increase over $m_i + \xi (\approx m_i)$ when $u_i(R_i)$ is plugged into utility maximization problem. So, b_i is set to a large value throughout the paper. When $b_i = 50$ and $c_i = 8$, the height $U_i'(R_i)$ is 2 and drops to zero above m_i. When $c_i = 16$, the shape is exactly the same as when $c_i = 8$ except that the height is 4, which is doubled. We will take advantage of these properties in developing scheduling and CAC algorithm. $U_i'(R_i)$ for $i \in E$ has the same property except for the part $R_i \geq m_i^\delta$.

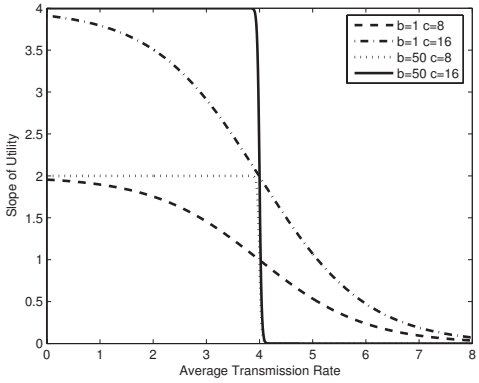


Fig. 6. $U_i'(R_i)$ with different values of b_i and c_i

Appendix II
Taylor Expansion of $U(R(t+1)) - U(R(t))$

By the definition of $U(R(t))$ and (2), we can write

$$\begin{aligned} &U(R(t+1)) - U(R(t)) \\ &= \sum_{i\in S} U_i(R_i(t+1)) - U_i(R_i(t)) \\ &= \sum_{i\in S} U_i\left(R_i(t) + \epsilon_t\left[r_{i,t+1}I_{i,t+1} - R_i(t)\right]\right) - U_i(R_i(t)). \end{aligned}$$

It follows from first order Taylor expansion in the neighborhood of $\epsilon_t = 0$ that

$$\begin{aligned} &U(R(t+1)) - U(R(t)) \\ &= \sum_{i\in S} U_i(R_i(t)) + \left[r_{i,t+1}I_{i,t+1} - R_i(t)\right] U_i'(R_i(t))\epsilon_t \\ &\quad + O(\epsilon_t^2) - U_i(R_i(t)). \\ &= \sum_{i\in S} \left[r_{i,t+1}I_{i,t+1} - R_i(t)\right] U_i'(R_i(t))\epsilon_t + O(\epsilon_t^2) \\ &= \sum_{i\in S} r_{i,t+1}I_{i,t+1}U_i'(R_i(t))\epsilon_t - \sum_{i\in S} R_i(t)U_i'(R_i(t))\epsilon_t \\ &\quad + O(\epsilon_t^2). \end{aligned}$$

Since $I_{i,t+1} \in \{0,1\}$ and $\sum_{i\in S} I_{i,t+1} = 1$, selecting the user having maximum $r_{i,t+1}U_i'(R_i(t))$ maximizes the first order term of $U(R(t+1)) - U(R(t))$. Eventually, such a selection will maximize $U(R(t))$ as $\epsilon_t \to 0$.

Appendix III
Proof of Theorem 3.1

Since the scheduler (7) maximizes the total utility, it is obvious that its limit point is the solution to the following

problem.

$$\begin{aligned} \max. \quad & \sum_i U_i(R_i) \\ \text{subject to} \quad & R \in \mathcal{F} \end{aligned} \tag{10}$$

where $\mathcal{F}$ is the feasible region of $R = [R_i, \forall i]$. By assumption, $\mathcal{F}$ is given as $\sum_i \frac{R_i}{\beta_i} \leq 1$ and $R \geq 0$. We will use the properties of the solution to problem (10) for all the proofs.

Let R^* be the optimal solution of (10), then, for user i and j, there can be three cases: $R_i^* > 0$ & $R_j^* > 0$, $R_i^* = 0$ & $R_j^* > 0$, and $R_i^* > 0$ & $R_j^* = 0$. We prove the theorem for each case by using an example of $\beta_i U_i'(R_i)$ and $\beta_j U_j'(R_j)$ satisfying the hypotheses of Theorem 3.1, which is shown in Fig. 7.

First, note that R^* exists at the boundary $\mathcal{F}^0$ of $\mathcal{F}$, i.e., $\sum_i \frac{R_i}{\beta_i} = 1$ because all the utility functions are strictly increasing and concave. Suppose that $R_i^* > 0$ and $R_j^* > 0$. If we shift a small amount $\eta\beta_i$ from R_i^* to R_j^*, then R_j^* will increase by $\eta\beta_j$ so that the new point still remains at $\mathcal{F}^0$. The change in the objective function by this shift is

$$-\eta\beta_i U_i'(R_i^*) + \eta\beta_j U_j'(R_j^*) \tag{11}$$

and this change must be non-positive due to the optimality of R^*. Similarly, shifting $\eta\beta_j$ from R_j^* to R_i^* yields the nonnegativity of (11). Consequently, we obtain

$$\beta_i U_i'(R_i^*) = \beta_j U_j'(R_j^*). \tag{12}$$

Applying (12) to the case shown in Fig. 7, we can assure that m_j cannot be fulfilled unless m_i is fulfilled. Since we are assuming that only one of m_i and m_j can be satisfied, user i will be guaranteed its minimum requirement (see the points marked by small black circles and **x**'s. Thus, user i has priority over user j. Suppose otherwise that $R_i^* = 0$ and $R_j^* > 0$, then it should hold $\beta_i U_i'(R_i^*) \leq \beta_j U_j'(R_j^*)$. However, the inequality cannot be satisfied for any R_i^* and R_j^* as seen in Fig. 7, and thus this case cannot happen (see the points marked by small black squares). For the opposite case where $R_i^* > 0$ and $R_j^* = 0$, it is easy to show that $R_i^* <= m_i$ and $R_j^* = 0$ and hence it is possible to guarantee m_i (see the points marked by ∇). Therefore, the theorem holds.

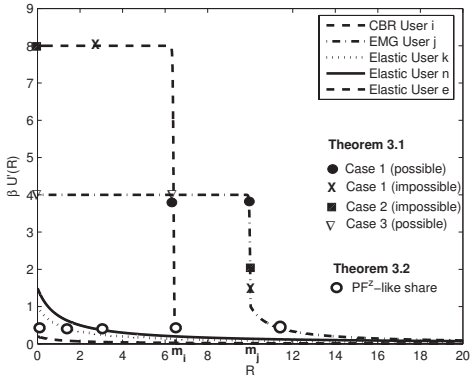


Fig. 7. $\beta U'(R)$ over R and pair of optimal solutions for each case

APPENDIX IV
PROOF OF THEOREM 3.2

By applying the similar argument used in the proof of Theorem 3.1, we can easily prove the theorem. First, if there is enough capacity so that some elastic users and elastic parts can get nonzero throughput, then the optimality condition (12) will hold for all the users achieving nonzero throughput. Thus, for example, the optimal point will be achieved as shown in Fig. 7 (5 points marked by small white circles). Notice that the elastic users have different values of $\beta U'(R)$ since we do not adjust their priorities. In the figure, it can be easily inferred that $\beta_n > \beta_k > \beta_e$.

Observe that CBR user j is exactly guaranteed its minimum requirement, and elastic users or elastic parts cannot achieve throughput before the requirement of CBR user j having highest priority is fulfilled. For the leftover capacity, the elastic user or part with better channel condition, i.e., larger β, yields higher throughput, but the users with extremely bad channel condition get no throughput. For example, elastic user n yields higher throughput than any other elastic user or elastic part and elastic user e gets no throughput. Therefore, the leftover capacity is shared by EMG users and the share is PF^z-like.

REFERENCES

[1] S. Shakkottai and A. L. Stolyar, "A study of scheduling algorithms for a mixture of real and non-real time data in hdr," Bell Laboratories, Lucent Technologies, Oct. 2000.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, and P. Whiting, "Cdma data qos scheduling on the forward link with variable channel conditions," Bell Laboratories, Lucent Technologies, Apr. 2000.

[3] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks Journal (Elsevier)*, vol. 41, no. 4, pp. 451–474, 2003.

[4] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *Proc. IEEE INFOCOM 2005*, Miami, Mar. 2005.

[5] W. Li, "Overview of fine granularity scalability in mpeg-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[6] Z. Cao and E. W. Zegura, "Utility max-min: An application-oriented bandwidth allocation scheme," in *Proc. IEEE INFOCOM 1999*, New York, Mar. 1999, pp. 793–801.

[7] F. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Sep. 1998.

[8] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of cdma-hdr: A high efficiency-high data rate personal communication wireless system," in *Proc. IEEE VTC2000-Spring*, Tokyo, May 2000.

[9] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Trans. Wireless Communications*, vol. 3, no. 4, pp. 1250–1259, Jul. 2004.

[10] ——, "Convergence of proportional-fair sharing algorithms: Extensions of the algorithm," *Brown University, Applied Math. LCDS Report*, 2003.

[11] S. Shenker, "Fundamental design issues for the future internet," *IEEE Journal on Selected Areas Communications*, vol. 13, no. 7, pp. 1176–1188, Sep. 1995.

[12] "Matlab simulation files." [Online]. Available: http://www.ece.utexas.edu/ iwong/OFDMAResAllocSim.htm