Dynamic Rate Control Algorithms for HDR Throughput Optimization

Sem Borst, Phil Whiting
Bell Laboratories, Lucent Technologies
P.O. Box 636, Murray Hill, NJ 07974

Abstract— **The relative delay tolerance of data applications, together with the bursty traffic characteristics, opens up the possibility for scheduling transmissions so as to optimize throughput. A particularly attractive approach, in fading environments, is to exploit the variations in the channel conditions, and transmit to the user with the currently 'best' channel. We show that the 'best' user may be identified as the maximum-rate user when the feasible rates are weighed with some appropriately determined coefficients. Interpreting the coefficients as shadow prices, or reward values, the optimal strategy may thus be viewed as a revenue-based policy.**

Calculating the optimal revenue vector directly is a formidable task, requiring detailed information on the channel statistics. Instead, we present adaptive algorithms for determining the optimal revenue vector on-line in an iterative fashion, without the need for explicit knowledge of the channel behavior. Starting from an arbitrary initial vector, the algorithms iteratively adjust the reward values to compensate for observed deviations from the target throughput ratios. The algorithms are validated through extensive numerical experiments. Besides verifying long-run convergence, we also examine the transient performance, in particular the rate of convergence to the optimal revenue vector. The results show that the target throughput ratios are tightly maintained, and that the algorithms are well able to track changes in the channel conditions or throughput targets.

Keywords— **Dynamic rate control, fading channels, Quality-of-Service, target throughput ratios, throughput optimization, varying channel conditions.**

I. INTRODUCTION

Next-generation wireless networks are expected to support a wide range of services, including high-rate data applications. In contrast to voice users, data applications can usually sustain some amount of packet delay, as long as the throughput over somewhat longer intervals is sufficient. The relative delay tolerance of data applications, together with the bursty traffic characteristics, opens up the potential for scheduling transmissions so as to optimize throughput. A coordinated approach along these lines is proposed in [3].

A related approach may be advocated for low-mobility scenarios such as indoor networks. In such environments, Rayleigh fading frequencies can be quite low, and the fading levels can even be anticipated to some extent. For example, fading can be measured by having the base station provide a pilot signal which can be measured by all the users. These measurements can be fed back to the base station, and used to estimate fading levels and hence user rates in subsequent slots. This is the approach proposed in Qualcomm's High Data Rate (HDR) scheme [4]. Clearly, it is then advantageous to exploit the variations in the feasible rates, and transmit to the currently 'best' user.

With a little simplification, let us suppose that at the start of each slot the base station has perfect knowledge of the maximum feasible rate at which each user can receive and decode a signal with some acceptably low error probability. The question then arises what the 'best' user is to be selected for transmission. We show that the 'best' user may be identified as the maximum-rate user when the feasible rates are weighed with some appropriately determined coefficients. Interpreting the coefficients as shadow prices, or reward values, the optimal strategy may thus be viewed as a revenue-based policy. We prove that revenue-based policies optimize throughput relative to pre-specified target ratios. These target values may be set arbitrarily, taking into account the Quality-of-Service requirements of the users, or possibly their current activity levels or locations.

Unfortunately, calculating the optimal revenue vector directly is a complicated problem, requiring detailed information on the channel statistics. Although the feasible rates of the users are assumed known slot by slot, the underlying probability distribution which is producing these rates is unknown. Even if it were known, it would not be easy to use, since the feasible rates might be dependent, so that the computations would be significantly hampered by the curse of dimensionality. To avoid these obstacles, we develop adaptive algorithms for determining the optimal revenue vector on-line in an iterative fashion, without the need for explicit knowledge of the channel behavior.

The application of these algorithms opens up two important possibilities to improve network performance. The first is that admission control can be applied by using a *probing technique*, an approach proposed in [2]. The second possibility is coordinated operation of the base stations in the network, which allows for load sharing and higher throughput for edge users.

The remainder of the paper is organized as follows. In Section II we present a detailed model description, and introduce a class of revenue-based scheduling strategies. We subsequently prove that revenue-based policies optimize throughput relative to pre-specified target ratios, for discrete rate distributions as well as for continuous rates in Sections III and IV, respectively. In Sections V, VI, and VII, we develop adaptive on-line algorithms for determining the optimal revenue vector in an iterative fashion. In Section VIII we describe some numerical experiments which we performed to examine the convergence properties of the proposed control algorithms. We make some concluding remarks in Section IX.

II. MODEL DESCRIPTION

We consider a base station serving data users. The base station transmits in slots of some fixed duration. In each slot, the base station transmits to exactly one of the users.

We assume that the feasible rates for the various users vary over time according to some stationary discrete-time stochastic process $\{(\ _1(\ ),\ldots,\ _M(\ )),\ =1,2,\ldots\}$, with $_m(\ )$ representing the feasible rate for user in the -th slot. We assume that the base station has perfect knowledge of the maximum fea-

sible rate $_m(\ )$ for user at the start of the -th slot, see also Remark II.1 below. Let $(\ _1,\ldots,\ _M)$ be a random vector with distribution the joint stationary distribution of the feasible rates.

We assume that the slot duration (1.67 ms in the HDR scheme) is relatively short compared to the relevant time scales in the traffic patterns and delay requirements of the data users. This opens up the possibility for scheduling the data transmissions so as to enhance performance. In particular, scheduling provides a potential mechanism for exploiting variations in the feasible rates so as to optimize throughput.

The data users may actually be thought of as the subset of active (backlogged) users among a greater population, which may change over time. For scheduling purposes, however, the separation of time scales allows us to think of the subset of active users as nearly static and continuously backlogged. (In practice, flow control algorithms such as TCP will typically be used to feed the data into the base station buffer at a relatively slow rate, comparable to the actual throughput provided to the user over the wireless link. Thus, the bulk of the backlogs will usually reside at the sender rather than the base station buffer.)

One of the most common performance objectives is throughput maximization. This can simply be achieved by assigning each slot to the user with the currently highest feasible rate. The disadvantage is that typically only a few strong users will ever be selected for transmission, causing starvation of all others.

To alleviate that problem, an alternative option is to equalize the (expected) throughput of the various users. This can easily be achieved by assigning each slot to the user with the currently smallest cumulative throughput. The downside is that this strategy does not exploit the variations in the feasible rates. Moreover, by insisting on equal throughput, a few weak users may cause the throughput of all others to be dramatically reduced.

In the present paper we assume there are throughput targets $_1,\ldots,\ _M$ defined for the various users. These target values may be set arbitrarily, taking into account the Quality-of-Service requirements of the users, or possibly their current activity levels or locations.

Denote $_m(\ ) := \ _m(\ )\ _m(\ )$, with $_m(\ )$ a 0–1 variable indicating whether or not the -th slot is assigned to user . Define $_m(\ ) := \mathbb{E}[\sum_{n=1}^{N}\ _m(\ )/\ ]$ as the expected average throughput received by user after slots.

We consider the problem of maximizing the minimum long-run expected average normalized throughput $:= \min_{m=1,\ldots,M}\ _m/\ _m$, with $_m := \liminf_{N\to\infty}\ _m(\ )$. The above problem may equivalently be formulated as maximizing subject to the constraint $\leq\ _m/\ _m$ for all $=1,\ldots,\ $. The constraints may in fact be sharpened to $=\ _m/\ _m$, since one can always reduce the throughput for the users with a surplus. With the equality constraints in place, the objective function may then be generalized to $\sum_{m=1}^{M}\ _m\ _m/\ _m$ for any positive vector $\in\mathbb{R}_+^M$.

In conclusion, the above-stated problem is equivalent to maximizing an arbitrary weighted sum of the throughputs, subject to the normalized throughputs being equal. Thus, the crucial observation is that any strategy which maximizes an arbitrary weighted combination of the throughputs, while equalizing the normalized throughputs, is optimal.

To formalize the above insight, we now introduce a class of revenue-based scheduling strategies. Suppose there were rewards $_1,\ldots,\ _M$ per bit transmitted to the various users. A revenue-based strategy assigns the -th transmission slot to the user $^*(\ )$ with the current maximum rate-reward product, i.e.,

$$^*(\ ) = \arg\max_{m=1,\ldots,M}\ _m\ \ _m(\ ).$$

Clearly, the above principle maximizes the revenue earned in each individual slot, and thus the total cumulative revenue, as well as the average revenue, hence the term revenue-based strategy. (It usually also matters exactly how ties are being broken. Regardless of the tie breaking rule, however, a revenue-based strategy will definitely *not* assign the -th slot to any user with $_k\ _k(\ ) < \max_{m=1,\ldots,M}\ _m\ _m(\ )$.) Now observe that revenue is simply a weighted combination of the throughputs. From our earlier observation, we thus conclude that any revenue-based policy which balances the throughputs, is in fact optimal, which provides the key principle underlying our further approach.

Finally, observe that setting throughput targets is equivalent to normalizing the feasible rates by the corresponding values. In the subsequent analysis, we therefore assume that the throughput targets are discounted for in the rates, and take $(\ _1,\ldots,\ _M) = (1,\ldots,1)$.

Remark II.1: In practice, there is always a small probability that a transmission fails because the signal cannot be successfully decoded. The results of the present paper then remain valid if $_m(\ )$ is redefined to represent the *expected* feasible rate, and the 0–1 variable $_m(\ )$ is amended to indicate both which user is selected and whether or not the transmission is successful. □

III. Discrete rate distribution

In this section we consider the case where the feasible rates $(\ _1,\ldots,\ _M)$ have a discrete distribution on some bounded set $\subseteq\mathbb{R}^M$. Since the feasible rates are assumed stationary, we restrict the attention to the class of stationary policies in order not to blur the presentation with technicalities. The analysis may readily be extended however to deal with non-stationary policies. We first introduce some notation. Let $_j$ be the stationary probability that the feasible rate vector is $\in$. We write $_{ij} = \ _i$ for $=(\ _1,\ldots,\ _M)\in$. Let $^\pi_{ij}$ be the probability that policy selects user for transmission when the feasible rate vector is $\in$. Then the minimum average throughput achieved under policy is $^\pi = \min_{i=1,\ldots,M}\ ^\pi_i$ with $^\pi_i = \sum_{j\in J}\ _j\ \ _{ij}\ ^\pi_{ij}$. Let ^{w} be the revenue-based strategy corresponding to the vector $=(\ _1,\ldots,\ _M)$. Without loss of generality, we assume that $\sum_{i=1}^{M}\ _i = 1$, since only the relative values of the revenues matter.

Lemma III.1: Policy is optimal iff $^\pi_{ij}$, $^\pi$ are an optimal solution to the following linear program:

$$\begin{aligned} &\max && \\ &\text{sub} && \leq\sum_{j\in J}\ _j\ \ _{ij}\ \ _{ij} \qquad =1,\ldots, \end{aligned} \tag{1}$$

$$\sum_{i=1}^{M} \quad _{ij} \le 1 \qquad \in$$
$$_{ij} \ge 0 \qquad = 1, \ldots, \quad , \quad \in \quad .$$

Proof

Let $^{*}_{ij}$, * be an optimal solution to the above linear program. Now consider the policy which assigns the slot to user with probability $^{*}_{ij}$ when the feasible rate vector is $\in$. The minimum average throughput achieved under this policy is $\min_{i=1,\ldots,M} \sum_{j \in J} \ _{j} \ _{ij} \ ^{*}_{ij} \ge \ ^{*}$. Thus, the optimal achievable throughput is at least *.

Conversely, for any policy , $^{\pi}_{ij}$, $^{\pi}$ are a feasible solution to the above linear program. Thus, the optimal achievable throughput is at most *, and hence exactly *. The statement then easily follows. □

The above lemma implies that for an optimal policy at most $| \ |+ \ -1$ of the variables $^{\pi}_{ij}$ are non-zero, which forces most of the variables to be one. Thus, only for a limited number of rate combinations the slots are shared among several users.

In Section II, we observed that a revenue-based policy which balances the throughputs is optimal. The next theorem shows that the revenue criterion is in fact a necessary optimality condition, in the sense that there exists a revenue vector * such that when user does *not* have the maximum rate-reward product i.e., $^{*}_{i} \ _{ij} < \max_{m=1,\ldots,M} \ ^{*}_{m} \ _{mj}$, then $^{\pi}_{ij} = 0$, i.e., user should *not* be selected for transmission. Thus, any optimal strategy *must* be a revenue-based policy associated with *, see [1] for a related stability result.

Theorem III.1: If policy is optimal, then there exists a vector $^{*} \ge 0$ such that for all $= 1, \ldots, \quad , \quad \in \quad ,$

$$^{\pi}_{ij} \left[\ ^{*}_{i} \ _{ij} - \max_{m=1,\ldots,M} \ ^{*}_{m} \ _{mj} \right] = 0. \qquad (2)$$

Proof

By Lemma III.1, the $^{\pi}_{ij}$ are an optimal solution to the linear program (1). Now let $^{*}_{i}$, $^{*}_{j}$ be an optimal solution to the dual problem of (1):

$$\min \quad \sum_{j \in J} \ _{j} \qquad (3)$$
$$\text{sub} \quad \sum_{i=1}^{M} \ _{i} \ge 1$$
$$_{j} \ge \ _{j} \ _{ij} \ _{i} \qquad = 1, \ldots, \quad , \quad \in$$
$$_{i} \ge 0 \qquad = 1, \ldots,$$
$$_{j} \ge 0 \qquad \in \quad .$$

Then the complementary slackness conditions imply $^{\pi}_{ij}[\ ^{*}_{j} - \ _{j} \ _{ij} \ ^{*}_{i}] = 0$, while optimality forces $^{*}_{j} = \ _{j} \max_{m=1,\ldots,M} \ ^{*}_{m} \ _{mj}$, yielding (2). □

The dual problem (3) may be interpreted as follows. The variable $^{*}_{j} = \ _{j} \max \ ^{*}_{m} \ _{mj}$ represents the revenue generated in state , so that the objective function measures the total expected earned revenue. Also, optimality implies $\sum_{i=1}^{M} \ ^{*}_{i} = 1$. Thus, the dual problem amounts to finding a revenue vector * which minimizes the total expected earned revenue, subject to the constraint $\sum_{i=1}^{M} \ ^{*}_{i} = 1$.

In conclusion, for policy $^{w^*}$ to balance the throughputs, the revenue vector * must minimize the total expected earned revenue, which may also be derived as follows. For any vector with $\sum_{i=1}^{M} \ _{i} = 1$, the total expected earned revenue is $(\ ) = \sum_{i=1}^{M} \ _{i} \ ^{\pi^{w}}_{i} \ge \sum_{i=1}^{M} \ _{i} \ ^{\pi^{w^*}}_{i} \ge \sum_{i=1}^{M} \ _{i} \min_{m=1,\ldots,M} \ ^{\pi^{w^*}}_{m} = \sum_{i=1}^{M} \ ^{*}_{i} \min_{m=1,\ldots,M} \ ^{\pi^{w^*}}_{m} = \sum_{i=1}^{M} \ ^{*}_{i} \ ^{\pi^{w^*}}_{i} = \ (\ ^{*})$.

IV. Continuous rate distribution

In this section we consider the case where the feasible rates $(\ _{1}, \ldots, \ _{M})$ have a continuous distribution on some bounded set $\subseteq \mathbb{R}^{M}$. We first introduce some notation. Let $(\ )$ be the stationary density of the feasible rate vector. We write $_{i}(\ ) = \ _{i}$ for $= (\ _{1}, \ldots, \ _{M}) \in \quad$. Let $^{\pi}_{i}(\ )$ be the probability that policy selects user for transmission when the feasible rate vector is $\in \quad$.

Lemma IV.1: Policy is optimal iff $^{\pi}_{i}(\ )$, $^{\pi}$ are an optimal solution to the following mathematical program:

$$\max \qquad (4)$$
$$\text{sub} \quad \le \int_{u \in U} \ (\ ) \ _{i}(\ ) \ _{i}(\ ) \qquad = 1, \ldots,$$
$$\sum_{i=1}^{M} \ _{i}(\ ) \le 1 \qquad \in$$
$$_{i}(\ ) \ge 0 \qquad = 1, \ldots, \quad , \quad \in \quad .$$

The proof of the above lemma is similar to that of Lemma III.1.

In Section II, we reasoned that a revenue-based policy which balances the throughputs is optimal. The next theorem shows that the revenue principle is in fact a necessary optimality criterion, in the sense that there exists a revenue vector * such that if user does *not* have the maximum rate-reward product on some set of non-zero measure, then user should not be selected for transmission on that set. Thus, in the above sense, any optimal strategy *must* be a revenue-based policy associated with *.

Theorem IV.1: If policy is optimal, then there exists a vector $^{*} \ge 0$ such that for all $= 1, \ldots, \quad ,$

$$\int_{u \in U} \ ^{\pi}_{i}(\ ) \left[\ ^{*}_{i} \ _{i} - \max_{m=1,\ldots,M} \ ^{*}_{m} \ _{m}(\ ) \right] \ (\ ) \quad = 0. \qquad (5)$$

The proof of the above theorem is similar to that of Theorem III.1. (Although strong duality does not directly apply, the complementary slackness properties may be derived via discretization.)

V. ADAPTIVE ALGORITHMS

In the previous two sections we concluded that revenue-based policies optimize throughput relative to pre-specified target values. However, calculating the optimal revenue vector directly is a complicated problem, requiring detailed information on the channel statistics in the form of the joint stationary distribution of the feasible rates $(\ _1, \ldots, \ _M)$. Instead, we develop adaptive scheduling algorithms for determining the optimal revenue vector on-line in an iterative fashion without the need for explicit knowledge of the channel behavior.

In the next two sections we assume that the channel state is governed by some discrete-time irreducible Markov chain with a finite discrete state space $\mathcal{S}$. When the channel state is $\in \mathcal{S}$, the feasible rates have some continuous -dimensional distribution $_s(\cdot)$ on $\mathcal{R} \subseteq [\ _{\min}, \ _{\max}]^M$, $0 < \ _{\min} < \ _{\max} < \infty$, with zero probability measure in any set of Lebesgue measure zero. In practice, the feasible rates will typically have to be selected from a limited set of discrete values. However, we may adhere to the above assumptions by simply adding a small random perturbation. By choosing the random perturbation sufficiently small, the true achieved throughputs should be arbitrarily close to the perturbed ones.

Denote by $\mathcal{W} := \{\ \in \mathbb{R}_+^M : \sum_{m=1}^{M} \ _m = 1\}$ the set of all price vectors. For any $\in \mathcal{W}$, denote by $\Xi_m(\ )$ the expected average throughput per slot received by user under price vector in stationarity. Define $\Xi_{ave}(\ ) := \frac{1}{M}\sum_{m=1}^{M} \Xi_m(\ )$, $\Xi_{\min}(\ ) := \min_{m=1,\ldots,M} \Xi_m(\ )$, and $\Xi_{\max}(\ ) := \max_{m=1,\ldots,M} \Xi_m(\ )$ as the average, the minimum, and the maximum expected throughput per slot under price vector over all users, respectively.

Denote by * the optimal revenue vector, i.e., the price vector which balances the expected throughputs. To facilitate the presentation, we assume that * is unique. The analysis may readily be modified for the case where there is a whole range of optimal price vectors.

VI. TWO USERS

We first focus on the case of two users. In the next section, we consider the situation with an arbitrary number of users.

Before describing the algorithm in detail, we first introduce some useful notation. With minor abuse of notation, we write $= \ _1$, so that $\ _2 = 1 - \ $. Denote $\Delta\ (\ ) := \ _1(\ ) - \ _2(\ )$, and define $(\ ) := \sum_{n=1}^{N} \Delta\ (\ )$ as the difference in cumulative throughput between users 1 and 2 after slots. The absolute difference $|\ (\ )|$ is referred to as the throughput *gap*. We say that the throughput gap *widens* in the -th slot if $|\ (\ )| > \max_{n=1,\ldots,N-1} |\ (\ )|$. User 1 is said to be *leading* if $(\ ) > 0$, and is referred to as *lagging* otherwise, and vice versa for user 2. We say that a *cross-over* occurs in the -th slot if the leading and lagging users exchange positions, which means that the throughput gap changes sign, i.e., $(\ )\ (\ - 1) < 0$.

The algorithm may now be described as follows. In every slot, the user with the maximum price-rate product, at the current price value, is selected for transmission. Thus, the -th slot is assigned to user 1 if $(\ )\ _1(\ ) > (1 - \ (\ ))\ _2(\ )$, and to user 2 otherwise (ties being broken arbitrarily).

To drive the price sequence $(\ )$ towards the optimal value *, the price is adjusted over time on the basis of the observed throughput realizations. As long as the throughput gap does *not* widen, the price is left unaltered. However, if the throughput gap *does* widen, then the price is changed in favor of the deficit user, thus at the expense of the surplus user. The price of the leading user is decreased by $_{k(n)}$, while the price of the lagging user is simultaneously increased by the same amount.

To ensure convergence, a *reset* is triggered at every cross-over. The step size $_{k(n)}$ is then reduced by incrementing $(\ )$, with $\{\ _k, \ = 1, 2, \ldots\}$ a pre-determined convergent sequence (e.g. $_{k+1} = \ _1\ ^k$ with < 1, or $_k = \ _1\ ^{-\beta}$ with > 1).

We now proceed to demonstrate convergence of the above-described algorithm. We first state an important assumption.

Assumption VI.1: (Large-Deviations Assumption)

Let $^N_m(\ ,\ )$ be a random variable representing the average throughput per slot obtained by user over a period of slots under price vector , given that the initial state of the Markov chain is . Given a price vector $\in \mathcal{W}$ and > 0, there exist numbers $^\xi_m(\ )$, $^\xi_m(\ ) > 0$ such that for any initial state

$$\mathbb{P}\{|\ ^N_m(\ ,\ ) - \Xi_m(\ )\ | > \ \} \leq \ ^\xi_m(\ )\ \mathrm{e}^{-D^\xi_m(w)N},$$

$= 1, 2$.

It may be verified that the above assumption is satisfied for the feasible-rate process described earlier.

The above assumption ensures almost-sure convergence to the optimal revenue vector as established in the next theorem.

Theorem VI.1: For the scheduling algorithm described above, the price sequence $(\ )$ converges to the optimal price * wp 1, and consequently the sequence $(\ )$ converges to the optimal value $^{\pi^{w^*}}$ wp 1.

In preparation for the proof of the above theorem we first present two lemmas.

Lemma VI.1: The price sequence $(\ )$ cannot get permanently trapped in either of the intervals $[0, \ ^* - \ ]$ or $[\ ^* + \ , 1]$.

Proof

We only prove the statement for the interval $[\ ^* + \ , 1]$. The statement for the interval $[0, \ ^* - \ ]$ follows from symmetry considerations.

The idea of the proof is as follows. As long as the price remains in favor of user 1, the throughput difference continues to have a positive drift, and will wander off to infinity. As a result, the price will keep decreasing in fixed steps, and will eventually turn negative, which is not possible.

For the formal proof details we refer to [5]. □

Lemma VI.2: The price sequence $(\ )$ cannot move from the interval $[0, \ ^* + \ ]$ to the interval $[\ ^* + 2\ , 1]$ infinitely often. Similarly, $(\ )$ cannot move from the interval $[\ ^* - \ , 1]$ to the interval $[0, \ ^* - 2\ ]$ infinitely often.

Proof

We only prove the first statement, The second one follows from symmetry considerations.

The idea of the proof is as follows. In order for the price sequence to move from the interval $[0, \ ^* + \ ]$ to the interval $[\ ^* + 2\ , 1]$, it must cross the interval $[\ ^* + \ , \ ^* + 2\ ]$ from left to right. For that to happen, the algorithm must make a number of -wrong moves. By an -wrong move, we mean that the price is increased while the current price is at least above the optimal value *. As will be shown below, the expected number of -wrong moves before a cross-over occurs is finite. However, as cross-overs occur, the step size will get smaller and smaller, and the required number of -wrong moves for the interval to be crossed will get larger and larger. As a result, it will eventually become increasingly unlikely for the interval to be crossed.

For the formal proof details we refer to [5]. □

Proof of Theorem VI.1

Lemma VI.1 implies that the sequence $(\ )$ spends infinitely many times in the interval $[\ ^* - \ , 1]$ wp 1. Lemma VI.2 shows that the sequence $(\ )$ returns only finitely many times from the interval $[\ ^* - \ , 1]$ to the interval $[0, \ ^* - 2\ ]$ wp 1. Combining these two statements, we find that the sequence $(\ )$ spends only finitely many times in the interval $[0, \ ^* - 2\ ]$ wp 1. Similarly, we have that the sequence $(\ )$ spends only finitely many times in the interval $[\ ^* + 2\ , 1]$ wp 1. Hence, for any > 0, the sequence $(\ )$ will eventually enter the interval $[\ ^* - 2\ , \ ^* + 2\ ]$ wp 1, to never leave it again. Thus, the sequence $(\ )$ converges to the optimal price * wp 1.

By continuity, the sequence $\mathbb{E}[\ _m(\ )]$ converges to $\Xi_m(\ )$, $= 1, 2$. The convergence of $(\ )$ then immediately follows. □

Remark VI.1: Some interesting related algorithms are proposed in [1], [7], [8], [9], [10], where queue lengths instead of rewards are used as weight factors. These algorithms provide throughput guarantees in terms of bounded expected queue lengths (if achievable) rather than target ratios. □

VII. Arbitrary number of users

We now turn to the situation with an arbitrary number of users. In principle, the algorithm described in the previous section for the case of two users may be extended to several users, although there are some subtleties involved in identifying a proper rule for when to trigger a reset.

Here we consider a related but somewhat different algorithm, which may be described as follows. The algorithm makes price updates based on sample periods of pre-determined ever increasing size. Thus, the price updates occur at pre-determined slots $(\ )$, instead of randomly determined slots as before, with $(\ ) := \ (\ +1) - \ (\ )$ the length of the -th sample period. In every slot of the -th sample period, the price vector $(\ )$ is used for selecting a user for transmission. (From now on we use to index sample periods, rather than transmission slots as before.)

To drive the price sequence $(\ )$ towards the optimal point *, the price is adjusted over time on the basis of the observed throughput realizations. The *direction* in which the price vector is modified at the -th update is determined by a random vector $(\ (\ ))$ based on the throughput obtained during the -th sample period when the price vector $(\ )$ is used. The *size* of the -th update is $(\ ) = \ _{k(n)}$, with $\{\ _k, \ = 1, 2, \ldots\}$ a pre-determined convergent sequence. Thus, at the $(\ +1)$-th update, the price vector is recursively determined as

$$(\ +1) = \ (\ ) - \ (\ )\ (\ (\ )).$$

To ensure convergence, the step size $(\ )$ is reduced by incrementing $(\ )$ every time a reset is triggered. Intuitively, resets should occur rarely far away from the optimal point *, but occur readily once the price vector is close to *.

It remains to specify the exact rules for (i) how to determine the update direction $(\ (\ ))$, and (ii) when to trigger a reset.

(i) For every user the empirical average throughput over the sample period is computed. The users are then partitioned into two groups: (a) those with above-average throughput; (b) those with below-average throughput. The prices of the above-average users are decreased, while the prices of the below-average users are increased. As the sample size grows, so that with high probability the empirical average throughputs line up with the true expected throughputs, this ensures that the price vector gets closer to the optimal point * in some appropriate sense, as will be shown later.

Formally, the procedure may be described as follows. Denote by $_m$ the throughput received by user during a particular sample period in which price vector is used. Define $_{ave} := \frac{1}{M}\sum_{m=1}^{M} \ _m$ as the average throughput over all users. Denote by $\Omega^- := \{\ : \ _m \leq \ _{ave}\}$ and $\Omega^+ := \{\ : \ _m > \ _{ave}\}$ the groups of below-average and strictly above-average users, respectively. Then the price update direction $(\ )$ is determined as

$$_i(\ ) \ = \ \frac{i}{\sum_{m\in\Omega^-} \ _m} \qquad \in \Omega^-, \tag{6}$$

$$_j(\ ) \ = \ \frac{- \ j}{\sum_{m\in\Omega^+} \ _m} \qquad \in \Omega^+. \tag{7}$$

Note that the price ratios within both Ω^- and Ω^+ are maintained. This ensures that the expected throughput of the below-average users increases, while the expected throughput of the above-average users decreases, as may be easily checked. In case $\Omega^+ = \emptyset$, the price vector is simply left unaltered.

Also note that the above price update cannot be applied in case the price values of some of the users in Ω^+ are zero. To prevent that situation from happening, the price process will be restricted to the set $\mathcal{W}_\nu := \{\ \in \mathcal{W} : \ _m \geq \ \text{for all} \ = 1, \ldots, \ \}$, with $:= \ _{\min}/(\ _{\min} + (\ -1)\ _{\max})$. It is easily verified that $^* \in \mathcal{W}_\nu$. In order to restrict the price process to the set $\mathcal{W}_\nu$, the update is truncated at the boundary if necessary.

(ii) To ensure convergence, a reset is triggered under the condition that every user has been a member of Ω^+ at least once during a consecutive sequence of updates. Once the reset has occurred, the next one is not triggered until every user has been a member of Ω^+ at least once again.

We now proceed to prove convergence of the above-described algorithm. We first discuss a few important assumptions.

Large-deviations assumption

As described above, the algorithm works by making price updates based on samples of ever increasing size. To ensure

convergence, we need that as the sample size grows, a 'correct' price update direction is selected with sufficiently high probability. Given a price vector $\in \mathcal{W}$, user is called -below-average (respectively, -above-average) if $\Xi_m(\ ) < \Xi_{ave}(\ ) -$ (respectively, $\Xi_m(\ ) > \Xi_{ave}(\ ) +$). We say that the price update direction is ' -right' if all the -below-average users belong to Ω^- and have their price increased, and all the -above-average users belong to Ω^+ and have their price decreased. This ensures that the price vector gets closer to the optimal point * in some appropriate sense, as will be shown later. Now remember that at each update, the prices of the empirical below-average users are increased, while the prices of the empirical above-average users are decreased. Thus, for the price update direction to be 'correct', it is critical that the empirical average throughputs line up with the true expected throughputs. This then motivates the following assumption.

Assumption VII.1: (Large-Deviations Assumption)

Let $^n_m(\ )$ be a random variable representing the average throughput per slot obtained by user over a period of $(\ )$ slots under price vector in stationarity. Given a price vector $\in \mathcal{W}$ and > 0, there exist a -neighborhood $^\xi_\zeta(\ )$ of and numbers $^\xi_m(\ )$, $^\xi_m(\ ) > 0$ such that

$$\mathbb{P}\{|\ \ ^n_m(\ ') - \Xi_m(\ )\ | > \ \} \leq \ ^\xi_m(\ )\, \mathrm{e}^{-D^\xi_m(w)L(n)},$$

for all $' \in \ ^\xi_\zeta(\ ), \quad = 1, \ldots, \ .$

We refer to [5] for a proof that the above assumption is satisfied for the feasible-rate process described earlier.

Boundary conditions

We further require that when a correct price direction is selected, the update cannot be truncated to an arbitrarily small size. The following assumption implies that if a correct price direction is chosen, then for small enough step size , the price sequence will stay away from the boundary.

Assumption VII.2: There exist positive constants $^* > 0$, > 0 such that for all price vectors $\in \mathcal{W}_\nu$, for any -right direction $(\ )$, and for any $\in (0, \ ^*)$,

$$+ \quad (\ ) \in \mathcal{W}_\nu.$$

To check that the above assumption is satisfied, it suffices to verify that extremely low prices cannot be decreased and that extremely high prices cannot be increased. First consider a user with a price $_i < \ _{\min}/(\ _{\min} + (\ \ - 1)\ _{\max})$. Then the throughput of user is zero, which means that the price of user is increased if the price direction is right. Similarly, the throughput of a user with a price $_j > \ _{\max}/(\ _{\min} + \ _{\max})$ is -above-average for some > 0, so that the price of user is decreased if the price direction is right.

Function $(\cdot)$

As indicated above, we also need that when a correct price update direction is selected, the price vector gets closer to the optimal point * by some definite amount. To measure distance from *, we introduce a function $(\cdot)$ which attains a unique minimum at *. Define $\Gamma_\epsilon := \{\ \in \mathcal{W} : \Xi_{\max}(\ ) - \Xi_{\min}(\ ) \leq \ \}$ as an ' -neighborhood' of *. The following assumption implies that if a correct price update direction is chosen, then outside Γ_ϵ the reduction in the value of $(\cdot)$ for small enough step size , is at least times some constant of proportionality .

Assumption VII.3: There exist positive constants $^* > 0$, > 0, > 0 such that for all price vectors $\notin \Gamma_\epsilon$, for any -right direction $(\ )$, and for any $\in (0, \ ^*)$,

$$(\ \ + \ \ (\ )) \leq \ \ (\ ) - \ \ .$$

We will consider two alternative choices for the function $(\cdot)$. The first one is

$$(\ ) := \Xi_{\max}(\ ) - \Xi_{\min}(\ ),$$

i.e., the maximum difference in expected throughput between any pair of users. By definition $(\ ^*) = 0$, and $(\ ) \geq 0$ for all $\neq \ ^*$, with strict inequality in case the optimal price vector * is unique.

The second function that we will consider is

$$(\ ) := \sum_{m=1}^{M} \ _m \Xi_m(\ ),$$

i.e., the total expected revenue earned. As found in Section III, the optimal price vector * minimizes that quantity over all vectors in the set $\mathcal{W}$, i.e., $(\ ^*) \leq \ (\ )$ for all $\in \mathcal{W}$, $\neq \ ^*$, with strict inequality in case * is unique.

We refer to [5] for a proof that Assumption VII.3 is indeed satisfied for the above two $(\cdot)$ functions.

In contrast to the first one, the second $(\cdot)$ function is also suitable to show that Assumption VII.3 is satisfied for various alternative options to select a price update direction, for example

$$_{i^*} \ = \ 1 - \ > 0 \qquad ^* = \arg \min_{m=1,\ldots,M} \ _m, \tag{8}$$

$$_{j^*} \ = \ -1 \qquad ^* = \arg \max_{m=1,\ldots,M} \ _m, \tag{9}$$

and $_k = \ _n/(\ \ - 2)$ for all $\neq \ ^*, \ ^*$, for $_n$ a given positive sequence with $\lim_{n\to\infty} \ _n = 0$. In the sequel this will be referred to as the 'Update-Extreme' algorithm, as opposed to the procedure described earlier which will be called the 'Move-to-Average' algorithm.

The next theorem establishes almost-sure convergence to the optimal revenue vector * for the Move-to-Average algorithm.

Theorem VII.1: The price sequence $(\ )$ converges to the optimal price vector * wp 1, and consequently the sequence $(\ )$ converges to the optimal value $^{\pi^{w^*}}$ wp 1.

The above theorem is proved in [5]. The proof for the Update-Extreme algorithm is mostly similar, except for a somewhat different notion of a correct price update direction.

Remark VII.1: In the present paper we focus on establishing almost-sure convergence to the optimal revenue vector *. This critically relies on the step sizes $\{\ _k, \ = 1, 2, \ldots\}$ being a convergent sequence. As an alternative, the step sizes may be kept fixed at some given value . We expect that the price sequence will then continue to oscillate around *, but with smaller amplitudes for smaller values of . Observe however that there is an inherent trade-off between the accuracy achieved on the one hand and the speed the convergence, and thus the responsiveness to changing conditions, on the other hand. The value of then may be used to find the right balance between these two conflicting objectives. □

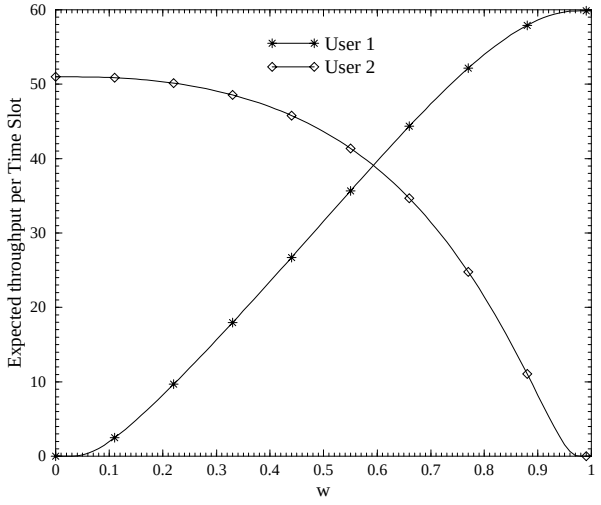

Fig. 1. Normalized expected throughput $\Xi_i(w)$ as function of w.

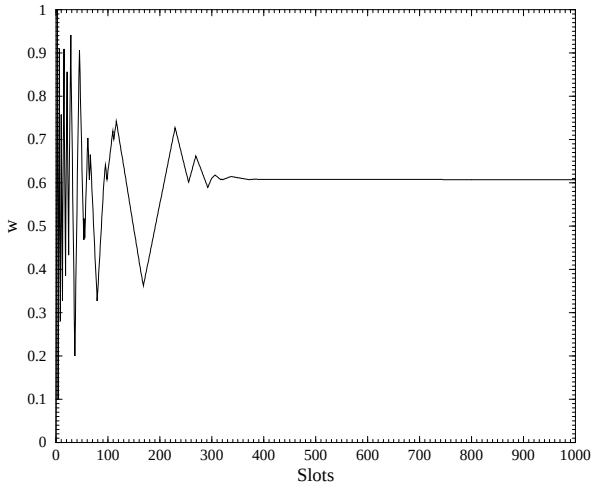

Fig. 2. Price trajectory for 2 users over 1000 slots.

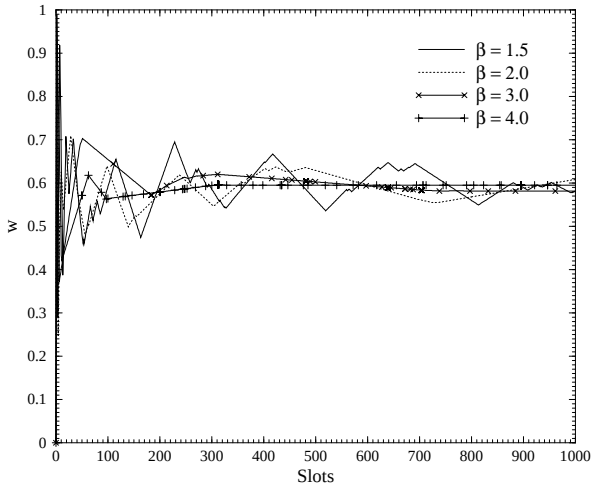

Fig. 3. Price trajectories for 2 users vs. w^* (non-geometric step sizes).

VIII. Numerical results

In this section we describe some numerical experiments which we conducted to investigate the convergence properties of the proposed control algorithms. Besides verifying long-run convergence, we also examine the transient performance, in particular the rate at which the prices converge to the optimal values.

In the first three experiments we consider continuous rate distributions. In the fourth experiment we assume a discrete distribution where the feasible rates are determined by a fading process via the signal-to-noise ratio. The fading process is modeled using a discrete number of sinusoidal oscillators as described by Jakes' model [6].

In the final three experiments, we examine how well the throughput ratios are maintained, and how well the algorithms are able to track changes in the channel conditions or throughput targets.

A. Two users with exponential rates

In the first experiment we consider a model of two users with independent rates. The feasible rate for user is governed by a conditional exponential distribution on some interval $[\ _{\min},\ _{\max}]$, i.e.,

$$_i(\ ) = \ _i^{-1}[1 - \mathrm{e}^{-\gamma_i(r - R_{\min})}], \qquad \in [\ _{\min},\ _{\max}],$$

with $\ _i = 1 - \mathrm{e}^{-\gamma_i(R_{\max} - R_{\min})}$ a normalization coefficient, $= 1, 2$. We take $[\ _{\min},\ _{\max}] = [10, 400]$ Kbits/s and assume $(\ _1,\ _2) = (0.02, 0.01)$. Thus, the feasible rate for user 2 is about twice as large in distribution as for user 1. The throughput target for user 2 is also set twice as large as for user 1, i.e., $(\ _1,\ _2) = (1, 2)$.

The normalized expected throughputs for these parameters as a function of are plotted in Figure 1. From the figure, we see that the optimal price is $^* \approx 0.6$, which may more precisely be determined as $^* \approx 0.593$ using bisection.

We ran the control algorithm described in Section VI for 1000 slots. We used step sizes $_{k+1} = \ ^k\ _1$, with initial value $\ _1 = 0.5$ and reduction factor $= 0.9$. The resulting price trajectories are graphed in Figure 2 for a period of 1000 slots. Observe that the prices converge to the optimal values in roughly 300 slots, which corresponds to about 0.3 seconds of operation.

We repeated the above experiment for non-geometric step sizes $_k = \ _1\ ^{-\beta}$, with successively chosen as 1.5, 2.0, 3.0, 4.0. Note that the sum of the price changes is still convergent, although the step sizes decay slower than before. The corresponding price trajectories are shown in Figure 3 for a period of 1000 slots. We see that convergence is considerably slower for smaller values of , i.e., slower decay of the step sizes.

B. Three users

In the second experiment we consider a scenario with three users. As before, the feasible rate for user follows a conditional exponential distribution on the interval [10, 400] with parameters $(\ _1,\ _2,\ _3) = (0.02, 0.01, 0.02)$. Thus, the feasible rate for user 2 is about twice as large in distribution as for users 1 and 3.

The target throughput ratios for the three users are set equal, i.e., $(\ _1,\ _2,\ _3) = (1, 1, 1)$. The optimal revenue vector is $^* \approx (0.424, 0.152, 0.424)$ as may be determined using numerical integration and two-dimensional bisection. Observe that the

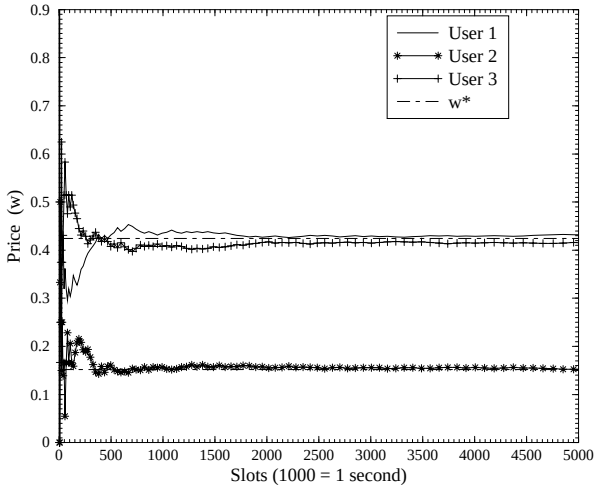

Fig. 4. Price trajectories for 3 users over 5000 slots vs. w^* (Move-to-Average algorithm).

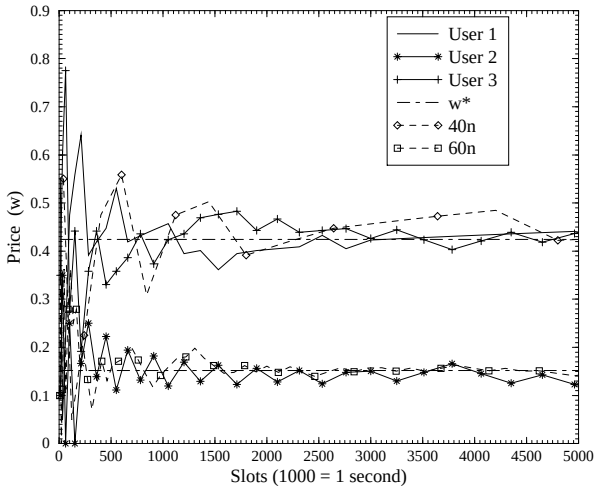

Fig. 5. Price trajectories for 3 users over 5000 slots vs. w^* (Update-Extreme algorithm).

optimal price for users 1 and 3 is higher than for user 1, as is required in order to obtain equal throughput, since the feasible rate for user 2 is stochastically larger.

We ran the two control algorithms described in Section VII for 5000 slots, or approximately 5 seconds of operation, with $(\ ) = 10$ slots for the -th update. This amounts to roughly 30 price updates. The initial revenue vector is set to $(1) = (0.3, 0.6, 0.1)$. We used step sizes $_k = {}^{-2}$, $= 1, 2, \ldots$. The resulting price trajectories are depicted as the solid curves in Figures 4 and 5. The revenue vector for the Update-Extreme algorithm after 30 price updates is $(30) \approx (0.441, 0.123, 0.436)$, quite close to the optimal one.

We repeated the above experiment for the Update-Extreme algorithm using 40 and 60 slots for the -th update, with the same power series for $_k$. The corresponding price trajectories are reproduced as the the dashed lines in Figure 5 for user 1 in the first case and user 2 in the second (with similar results for the remaining prices.) As expected, we see that using fewer samples per price update leads to a slower and 'noisier' convergence to

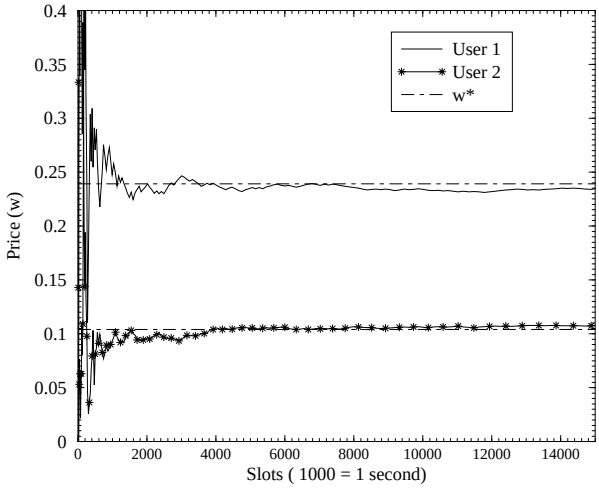

Fig. 6. Price trajectories for 8 users with 15,000 slots vs. w^* (Move-to-Average algorithm).

TABLE I
FEASIBLE RATE PER SLOT AS FUNCTION OF SNR.

Signal-to-Noise Ratio (dB)	Rate (bits)
$-5.0 <$ SNR	1000
$-10.0 <$ SNR ≤ -5.0	500
$-20.0 <$ SNR ≤ -10.0	250
$-30.0 <$ SNR ≤ -20.0	100
SNR ≤ -30.0	30

the optimal revenue vector *.

C. Eight users

In the third experiment we consider a situation with eight users. As before, the feasible rate for user follows a conditional exponential distribution on the interval [10, 400]. The exponents were chosen at random uniformly in [0.01, 0.05], and turned out to be approximately (0.0489, 0.0263, 0.0139, 0.0480, 0.0220, 0.0107, 0.0461, 0.0128).

The target throughput ratios are again set equal for all users. As before, we expect that a larger value of the exponent , inducing smaller feasible rates, requires a higher price in order to obtain equal throughput.

We ran the two control algorithms described in Section VII for 15,000 slots, or approximately 15 seconds of operation, with $(\ ) = 30$ slots for the -th update. This amounts to roughly 55 price updates. The initial revenue vector is set at random. We used step sizes $_k = {}^{-2}$, $= 1, 2, \ldots$. The resulting price trajectory for the Move-to-Average algorithm is graphed in Figure 6.

D. Discrete rates driven by a fading process

We now consider a case with discrete rates governed by independent fading processes as described by Jakes' model [6]. The mean received powers of user 1, 2 and 3 are -15.0 dB, 0.0 dB, and -10.0 dB, respectively. The feasible rates per slot then follow from Table 1.

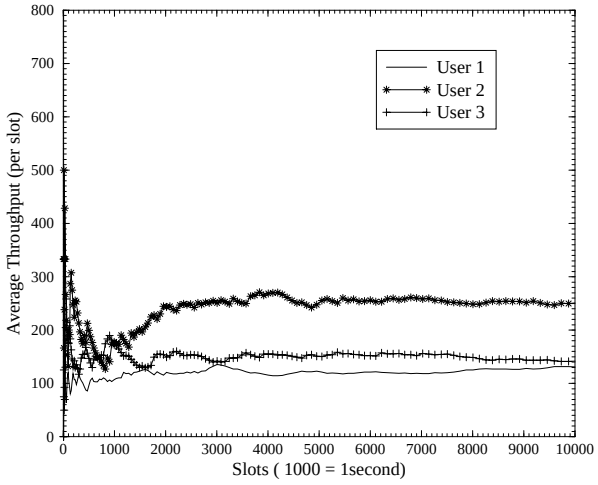

Fig. 7. Empirical average throughput for 3 users over 10,000 slots (Move-to-Average algorithm with $\delta_k = k^{-3/2}$).

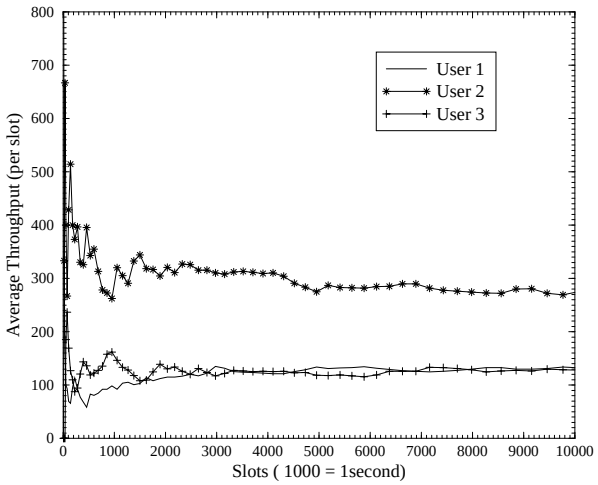

Fig. 8. Empirical average throughput for 3 users over 10,000 slots (Update-Extreme algorithm with $\delta_k = k^{-2}$).

The throughput target for user 2 is set twice as large as for users 1 and 3, i.e., $(\ _1, \ _2, \ _3) = (1, 2, 1)$.

We ran the two control algorithms described in Section VII for 10,000 slots, with $(\ ) =$ slots for the -th update. We used step sizes $_k = \ ^{-3/2}$ and $_k = \ ^{-2}$, $= 1, 2, \ldots$.

As explained earlier, the discrete rate values are perturbed by adding a small uniformly distributed random variable to obtain a continuous version of the problem. We thus ensure that the optimal control algorithm is determined by the revenue vector only.

The empirical average throughputs are depicted in Figures 7, and 8. The achieved throughputs under the Update-Extreme algorithm are approximately 130 bits per slot for both users 1 and 3 and 270 bits per slot for user 2, quite close to the target ratios. Under the Move-to-Average algorithm the realized throughputs are reasonably close to the target ratios too, provided the step size is reduced sufficiently slowly as in Figure 7.

The corresponding price trajectories are displayed in Figures 9, and 10. We see that that under the Update-Extreme algorithm the prices converge to the optimal values in about 5 seconds. Under the Move-to-Average algorithm the prices converge fairly quickly too, unless the step size is reduced so fast that the process gets essentially overdamped.

E. Comparison with a forcing scheme

We now compare the revenue-based algorithms with a forcing scheme. The forcing scheme assigns the -th transmission slot to the user $^*(\ )$ with the current minimum normalized throughput, i.e.,

$$^*(\ ) = \arg \min_{m=1,\ldots,M} \ _m(\ )/ \ _m.$$

By construction, the forcing scheme realizes the target throughput ratios perfectly, in the sense that with probability 1,

$$\frac{_i(\ )}{_j(\ )} \to \frac{_i}{_j}, \qquad \text{as} \quad \to \infty$$

for all pairs of users , $= 1, \ldots,$.

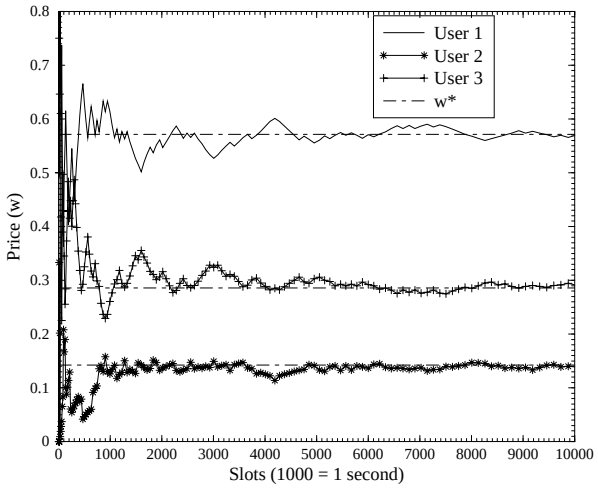

Fig. 9. Price trajectories for 3 users over 10,000 slots (Move-to-Average algorithm with $\delta_k = k^{-3/2}$).

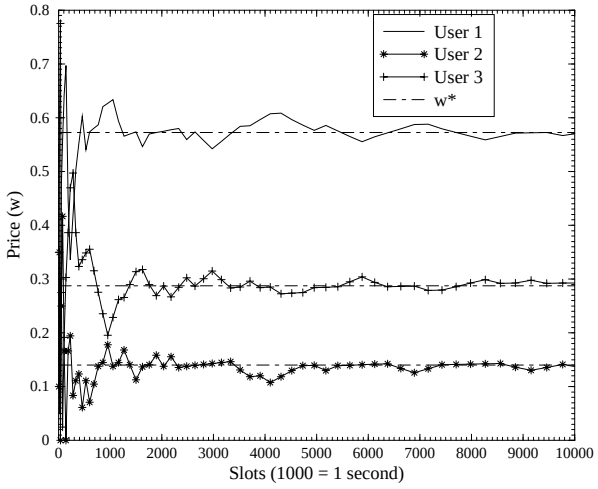

Fig. 10. Price trajectories for 3 users over 10,000 slots (Update-Extreme algorithm with $\delta_k = k^{-2}$).

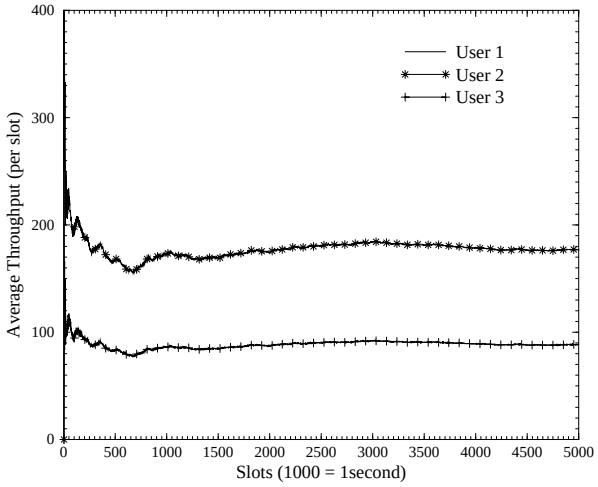

Fig. 11. Empirical average throughput for 3 users over 5000 slots (forcing algorithm).

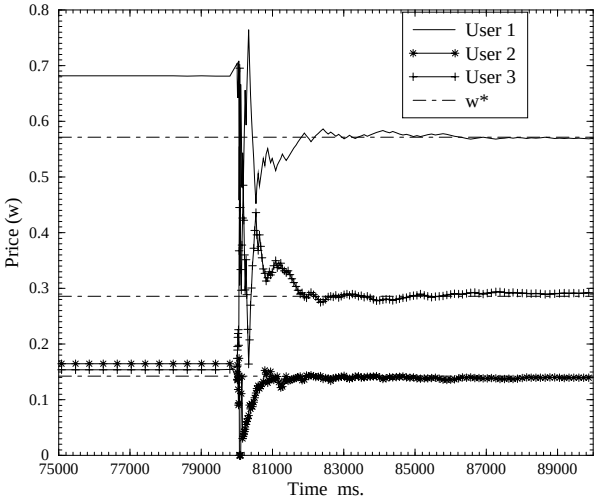

Fig. 12. Price adjustment to allow for data burst for user 3 (Move-to-Average algorithm).

The downside of the forcing scheme of course, is that it generally achieves lower throughput in absolute terms, as it does not take advantage of the variations in the feasible rates. Under i.i.d. assumptions, the throughput obtained under the forcing scheme may in fact be computed in closed form as

$$_i(\quad) \rightarrow \quad _i \quad , \qquad \text{as} \qquad \rightarrow \infty.$$

with $\quad ^{-1} = \sum_{j=1}^{M} \quad _j / \mathbb{E}[\quad _j]$.

We repeated the experiment of the previous subsection for the forcing scheme. The empirical average throughputs are reproduced in Figure 11 for a period of 5000 slots. The achieved throughputs are approximately 90 bits per slot for both users 1 and 3, and 180 bits per slot for user 2. The results show how tightly the target throughput ratios are maintained under the forcing scheme. In absolute terms however, the throughput for all users is about 30% smaller than for the revenue-based algorithms.

F. Tracking capability

We now examine how well the algorithms are able to track sudden changes in the target throughput ratios or channel conditions. In the first experiment, the throughput target for user 3 is initially set to some low value. After 80 seconds, the throughput target is suddenly incremented to allow for the transmission of a data burst for user 3.

The resulting price trajectory for the Move-to-Average algorithm is plotted in Figure 12. The optimal price values for the new throughput ratios are also indicated as dashed straight lines. The results show that after a few oscillations the prices quickly settle down to the new optimal values.

IX. Conclusion

We considered the problem of scheduling data users with varying channel conditions so as to obtain the optimal long-run throughputs for given target ratios. We have shown that the problem may be solved by selecting users for transmission according to an optimal revenue vector * which balances the expected throughputs. We presented a wide class of stochastic control algorithms which ensure almost-sure convergence to * and thus achieve the optimal long-run throughputs.

Numerical experiments showed that the convergence to the optimal revenue vector is in practice quite rapid (of the order of a few seconds), and that the algorithms have the ability to track changes in the channel conditions and throughput targets. Further experiments are required to determine which form of the algorithm is most adequate for implementation in the HDR scheme. The algorithms may also be enhanced by allowing the step sizes or the sample sizes to be adapted in response to non-stationary changes in the feasible rate declarations

References

[1] Andrews, D.M., Kumaran, K., Ramanan, K., Stolyar, A.L., Vijayakumar, R., Whiting, P.A. (2000). CDMA data QoS scheduling on the forward link with variable channel conditions. Technical Memorandum, Bell Laboratories, Lucent Technologies.

[2] Bambos, N., Chen, S.C., Mitra, D. (1995). Channel probing for distributed access control in wireless communication networks. In: *Proc. IEEE Globecom '95*.

[3] Bedekar, A., Borst, S.C., Ramanan, K., Whiting, P.A., Yeh, M. (1999). Downlink scheduling in CDMA data networks. In: *Proc. IEEE Globecom '99*, 2653–2657.

[4] Bender, P., Black, P., Grob, M., Padovani, R., Sindhushayana, N., Viterbi, A. (2000). CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users. *IEEE Commun. Mag.* **38**, 70–77.

[5] Borst, S.C., Whiting, P.A. (2000). Dynamic rate control algorithms for HDR throughput optimization. Technical Memorandom 10009626-000828-21TM, Bell Laboratories, Lucent Technologies. Submitted for publication.

[6] Jakes, W.C. (1974). Multipath interference. In: *Microwave Mobile Communications*. Ed. W.C. Jakes, IEEE Press, Piscataway.

[7] Kahale, N., Wright, P.E. (1997). Dynamic global packet routing in wireless networks. In: *Proc. IEEE Infocom '97*, 1441–1421.

[8] Shakkottai, S., Stolyar, A.L. (2000). A study of scheduling algorithms for a mixture of real and non-real time data in HDR. Technical Memorandum, Bell Laboratories, Lucent Technologies.

[9] Shakkottai, S., Stolyar, A.L. (2000). Throughput-optimal scheduling for time-varying channels: the exponential rule. Technical Memorandum, Bell Laboratories, Lucent Technologies.

[10] Tassiulas, L. (1998). Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In: *Proc. IEEE Infocom '98*, 553–559.