Managing Resources and Quality of Service in Heterogeneous Wireless Systems Exploiting Opportunism

Shailesh Patil and Gustavo de Veciana
{patil, gustavo}@ece.utexas.edu

***Abstract*— We propose a novel class of opportunistic scheduling disciplines to handle mixes of real-time and best effort traffic at a wireless access point. The objective is to support probabilistic service rate guarantees to real-time sessions while still achieving opportunistic throughput gains across users and traffic types. We are able to show a 'tight' stochastic lower bound on the service a real-time session would receive assuming that the users possibly heterogenous capacity variations are known or estimated, and are fast fading across slots. Such bounds are critical to enabling predictable quality of service and thus the development of complementary resource management and admission control strategies. Our simulation results show that the scheme can achieve more than 90% of the maximum system throughput capacity while satisfying the QoS requirements for real-time traffic, and that the degradation in system throughput is slow in the number of real-time users, i.e., inter and intra class opportunism are being properly exploited. We note however, that there is a tradeoff between strictness of QoS requirements and the overall system throughput one can achieve. Thus if QoS requirements on real-time traffic are very tight, one would need to simply give priority to real-time traffic at a loss throughput derived from opportunism.**

I. Introduction

Motivation. Wireless networks are moving towards providing broadband services. These services will support a mixture of real-time streams (e.g., video/voice and multimedia) and best effort data transfers (like file downloads or web browsing). From a user's perspective this requires a scheduling scheme which can ensure quality of service (QoS) to a real-time session and/or minimize transfer delays associated with best effort sessions (see Figure 1). From a system perspective one would like the capability to admit a large number of real-time sessions while, at the same time, maximize revenue generating data throughput. To manage such traffic mixes on limited wireless resources, one must be able to predict and evaluate the likelihood that QoS commitments can be met, i.e., devise complementary resource allocation and call admission strategies.

At the same time a key feature of wireless systems relative to the traditional wireline systems is that the channel capacity, or service rate, may exhibit temporal variations. This allows one to consider scheduling policies that choose to send to, or receive from, the user(s) which at a given time has(have) the 'best', e.g., highest channel capacity. Such 'opportunistic scheduling' can lead to good increases in the aggregate throughput of a wireless system. Devising an opportunistic scheduling scheme that handles mixes of traffic while permitting some degree of performance prediction is the objective of this paper.

Challenges. In meeting this objective one must overcome somewhat formidable challenges. End system device diversity, space-time variations in the propagation environment, interference, and different user mobility patterns, lead to heterogeneity and variability in the channel capacity a user would see. Indeed, those which are near a base station would generally experience a much better average channel capacity than those at the edge of a coverage area. Further users will undergo fast fading, i.e., short term variations that depend on their location, and whose statistics may vary over time e.g., time varying for e.g. mobile users. Of course the idea is to exploit such fluctuations through an opportunistic scheduler, yet one must do this with care to avoid starving users whose channel characteristics are less favorable. At the same time real-time and best effort sessions will have different traffic load statistics and heterogenous QoS requirements which a scheduler should somehow address. Finding a practical approach to opportunistic scheduling that harmoniously deals with the above discussed factors and at the same time allows 'prediction' towards supporting quality of service is the challenge we face.

Related work. Perhaps the first opportunistic scheduling scheme was proposed in [10], here a greedy strategy known as *max rate* scheduling was introduced, i.e., sending to a user whose channel capacity is currently the highest. Max rate scheduling achieves high overall system throughput, but if users have heterogenous channel capacity distributions, it neglects users with poorer channels. Several approaches have been proposed to address both unfairness and performance issues. The best known is *proportionally fair* scheduling [7], [19], and subsequently, among others, *modified-largest weighted delay first* [1] and *exponential rule* [18] where proposed. These mechanisms try to achieve multiple objectives of ensuring QoS, maximizing throughput while achieving 'fairness', etc, and succeed to various degrees. In [17] the performance of these three scheduling algorithms was compared from the perspective of providing QoS guarantees and the exponential rule was found to be best. More recently Maximum Throughput with Minimum/Maximum Rate and Proportionally Fair with Minimum/Maximum Rate have been proposed to satisfy users' QoS guarantees in [2], the idea there is to weight a user's current rate by a factor based on how well the user is doing relative to its target rate.

The above mentioned schemes achieve multiple objectives by attaching priority weights to users and choose to serve the user with the highest weighted channel capacity. These weights can be a function of service a user has seen to date, the present queue backlog, and QoS or fairness requirements among users. The flexibility in assigning these weights allows one to handle heterogeneity in channel capacity distributions. However proper selection of these weights is very difficult, because they will in general be jointly dependent on the channel capacity distributions and traffic characteristics of all users, and may be impossible for dynamic systems where the activity levels and numbers of users vary. As such it is unclear whether a meaningful performance prediction, resource management and call admission policies could be devised based on such schedulers.

Another class of opportunistic scheduling approaches assumes the channel distributions of users are known or estimated [11][12][4][15]. The idea is to schedule a user whose current rate is least likely relative to his current channel distribution, i.e., in the highest quantile – we shall refer to these as *maximum quantile scheduling*. This approach has several desirable properties in terms of handling heterogeneity, decoupling users performance, and permitting prediction of *long term* average throughput. In our own work [14] we have shown that in practice this scheme provides excellent throughput, packet delay and best effort flow delay performance even when distributions need to be estimated on the fly. However alone this approach can not address short term QoS guarantees required for real-time sessions. Nevertheless it will serve as the building block for the work in this paper.

The work of [21][22][20] suggests realizing QoS guarantees based on an effective bandwidth concept. An evaluation of the offered QoS is based on determining the effective capacity which requires knowledge or estimation asymptotic log moment generating function of the channel capacity process seen by a user at the base station. The approach initially focussed on the case where all users have homogeneous channel capacity distributions which is unlikely in practice. Extension to the case where users have heterogeneous channel capacity distributions was discussed in [20]. However the extension required evaluating a complicated function over a continuous range of parameters for each user, making the scheme hard to implement. Furthermore, because the underlying analysis is based on large buffer large deviations, the resulting QoS estimate may not be relevant on the short time scales relevant for real-time users. The shortcomings of this work highlights some of the difficulties we mentioned earlier. However, note that if we are to predictably ensure QoS guarantees it is likely that the knowledge of users' channel capacity statistics at the base station will be required.

There is very little work on opportunistic scheduling and the integration of real-time and best effort traffic. A simple solution may be to give absolute priority to real-time over best effort traffic. If the real-time sessions were scheduled opportunistically then such scheme would enable one to exploit 'intra class opportunism', i.e., opportunism among users of the same class. Yet due to the coupling among real time streams it is unclear, how performance could be predicted. Furthermore, ideally one would like to also exploit 'inter class opportunism', i.e., opportunism from both the real-time and the best effort users competing for service.

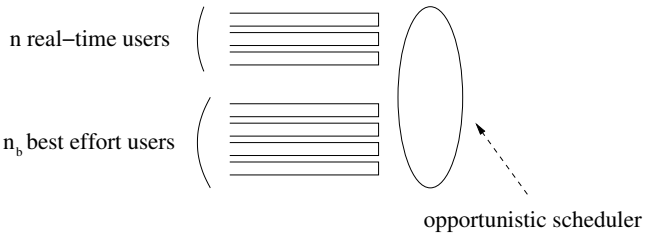


Fig. 1. Scheduling a mixture of real-time and best effort users from a wireless base station.

Contributions and organization. In this paper we propose a novel opportunistic scheduling mechanism and resource allocation strategy that fulfil multiple objectives. Under the assumption that the (possibly heterogenous) channel capacity distributions of users are known (or estimated) at the base station and stay stable for moderate timescales, we are able to ensure probabilistic guarantees on the rate experienced by real-time sessions over short time scales. For simplicity, we begin in Section II by considering the case where sessions see independent, identically distributed channel capacity variations, i.e., homogeneous channel characteristics and want the same QoS guarantees. Under fast fading, we develop stochastic lower bounds for the service received by real-time users which can be used as a basis for making admission control and resource allocation decisions. This bound is significant because it allows resource allocation decision for a real-time user to be independent of other users' channel capacity distributions. This independence holds true even when users have heterogenous channel capacity variations and QoS requirements, which is considered in Section III. *Therefore the proposed opportunistic scheduler can predictably guarantee QoS over short time scales while still benefiting from opportunism when users have heterogeneous channel capacity distributions, i.e., exploiting both intra class and inter class opportunism.* This is verified in the simulation results presented in Section IV which show that we can satisfy strong QoS guarantees while achieving more than 90% of the system throughput realized under max rate scheduling. This is excellent since for a static saturated set of users, max rate maximizes the system throughput. Our analysis assumes users channels are fast fading, i.e., i.i.d., across slots, but we propose a heuristic modification that would make the scheduler robust if in fact users capacity variations were dominated by a slow fading environment. This claim is again supported by our simulations. Section V concludes the paper.

II. Scheduling and Resource Allocation for Symmetrical Channel Capacity Distributions

A. System Model and Notation

For simplicity, we consider only downlink scheduling from a base station to multiple users (the scheme can be applied for uplink scheduling as well). We divide time into equal sized slots with at most one user served per slot. During each slot, each user feeds back the channel capacity or rate (we will use

these terms interchangeably) it can support to the base station which in turn makes a decision on which user to serve. Such a system model is used in CDMA-HDR systems defined in the CDMA2000 IS-856 standard [7].

Channel assumptions. In practice, characterizing the channel capacity or rate seen by a user is quite complicated. There are several factors that affect the capacity, they can be broadly classified into two classes[16]. First, there is large scale path loss in the 'average' capacity seen by a user due to the distance of a user from the base station, and the shadowing due to obstacles in the path between the user and the base station. Secondly, there is small-scale variation or fading in the instantaneous capacity due to multipath time delay spread, the speed of such variations depends on the Doppler spread seen by the user. Therefore a simple yet reasonably accurate model may be to view the channel capacity seen by a user as a quasi stationary random process, with the marginal distribution that changes following changes in the large scale path loss. These marginal distributions are likely to be different across users.

Note that ensuring QoS requires giving guarantees towards future service, this can be done only if the users' capacity distribution or some function of it is known or predicted at the base station. Also note that if a user's distribution changes, the guarantee given may or may not hold, thus one must constantly track and learn the distribution (or some function of it). If the channel is quasi-stationary on time scales where users' rate distribution estimates may be made reliably, then the base station can track and allocate resources as needed to ensure QoS goals are met. Of course, if users' capacity distribution is changing too fast, then it is virtually impossible to provide any kind of guarantee.

In this paper we will assume such quasi-stationary characteristics for users' channel capacities, and for analysis purposes assume the regime where the users' channels are in fact stationary. This will allow us to establish the resource requirements for each 'stationary' regime the user experiences:

Assumption 2.1: We assume the channel capacity (rate) for each user is a stationary ergodic process and these processes are independent, identically distributed (i.i.d.) across users. The channel capacity for each user is fast fading, i.e., the channel capacity for each user is independent across slots and remains constant during a slot. Further we assume that the marginal distribution for each user is either known a priori, or estimated by the base station.

Discussion of the assumptions. Let us first discuss the assumption that the base station knows, and in particular can estimate, the marginal distributions of the channel capacity processes. This can be achieved using simple book keeping on the users' feedback of the currently achievable rate, i.e., tracking. We need to know users' channel distribution for maximum quantile scheduling (for asymmetrical channel capacity distributions), and to perform resource allocation. We show in [14] that the throughput penalty due to estimation errors in users' channel distribution is not high for maximum quantile scheduling. This result will be informally stated later in this paper. Additionally as will be discussed in the final analysis, the base station only needs to know the mean and variance of a certain quantity users' channel distribution to perform resource allocation. In summary, the assumption is required, but is not unreasonable in a practical system - which tracks quasi-stationary changes in channel statistics.

Let us discuss the other assumptions. First as discussed earlier the channel capacities seen by users might indeed be roughly stationary over a reasonable period of time, particularly if users are at fixed locations. We conjecture based on the performance of maximum quantile scheduling under estimated rate distributions that the users' channel should be stationary for roughly $O(n_{tot}^2)$ number of time slots (here n_{tot} is the total number of users in the system) to allow us to collect sufficient samples for performing resource allocation and scheduling without penalizing throughput performance too much [14]. The assumption that users' rates are independent is also likely to be true, though a notable exception is the case where mobile users move in a correlated manner, e.g., along a highway. The assumption that users' channels are identically distributed is simplistic, this will be relaxed later when we incorporate heterogeneous channel capacities in our framework. For the channel capacity to be independent across slots, the channel must be fast fading with a coherence time equal to that of a slot's time period. This may not be realistic, but 'opportunistic beamforming' [19] can provide sufficient variability in the rates experienced by users across slots to roughly achieve this. Also, later we will propose a heuristic for the case where users' rates are correlated across slots. Note that we do not assume any specific channel model, this allows the scheme to work under any fading process users might be experiencing.

Notation. We begin by introducing some notation relevant to this section. For simplicity, the time period of a slot is fixed to a single time unit. Let $\mathbf{X}^i = (X^i(t) : t \in \mathbb{N})$ be a discrete time random process capturing the channel rate process of user i. By Assumption 2.1, $\mathbf{X}^i$'s are stationary and ergodic processes. Let X^i be a random variable representing the marginal distribution of $\mathbf{X}^i$. Again by Assumption 2.1, the channel is fast fading, therefore X^i captures the rate distribution seen by user i in a typical slot. Let $x^i(t)$ denote the realization of the channel capacity of user i for time slot t. According to Assumption 2.1, the base station knows the distribution of the X^i in addition to $x^i(t)$ for each user. Also since for now, we assume that the channel capacity distributions are i.i.d. across users , therefore we will sometimes drop the users' index in this section and denote X^i by X, a random variable whose distribution is same as that of the channel capacity of any of the users in the system.

Let $A_r(t)$ denote the set of active real-time users at time slot t, i.e., if $i \in A_r(t)$ the base station will allow user i to compete for the slot t. Note that the scheduling discipline will be responsible for deciding which real-time users are 'allowed' to contend for a slot. Also note that for convenience it is possible for an active real-time user in a slot to have no backlogged data. The set of active best effort users is denoted by $A_b(t)$. A best effort user j is said to be active only if it has a backlog prior to that slot. The set of active best effort users is denoted by $A_b(t)$ and define $A(t) := A_r(t) \cup A_b(t)$.

Under max rate scheduling, the base station receives channel capacity feedback $x^i(t)$ from each user in $A(t)$ and chooses the 'best' to serve. More formally, the access station chooses to serve user i during slot t if $x^i(t) = \max_{j \in A(t)} x^j(t)$. We let $X^{(l)} = \max\{X_1, ..., X_l\}$, where all X_j's are i.i.d. and $X_j \sim X$, i.e., $X^{(l)}$ is the maximum of l i.i.d. random variables. Consider a slot where user i is competing with $l-1$ other users. Conditioning on user i being selected for service, his conditional rate distribution X^i is the same as $X^{(l)}$. This follows easily by symmetry among the contending users. Let us discuss some properties of $X^{(l)}$ which will be useful in the proofs given later. $X^{(l)}$ are stochastically increasing in l, i.e., $\forall x$, $\Pr(X^{(l+1)} \geq x) \geq \Pr(X^{(l)} \geq x)$. This is usually denoted as $X^{(l+1)} \geq^{st} X^{(l)}$, and means that the probability $X^{(l)}$ takes a high value increases in l.

We shall let n denote the total number of real-time users and n_b the total number of best effort users (see Figure 1). For simplicity we assume that a user initiates only one type of session at a time, with exactly one real-time stream per real-time user, i.e., the number of real-time users is equal to the number of real-time streams.

QoS definition. The notion of QoS considered in this paper involves ensuring a user i sees a desired rate r over a frame of length τ with an outage probability of δ. More formally, we divide time into equal sized 'frames' of τ units and our goal is to ensure that for each of these frames

$$\Pr(S_i(\tau) > r\tau) \geq 1 - \delta,$$

where $S_i(\tau)$ is a random variable denoting the cumulative potential service to user i during a frame. For simplicity we restrict τ to take only integral values with respect to the time unit, i.e., the QoS guarantees are given only over an integral number of time slots.

If the traffic load of user i does not exceed $r\tau$ over a given frame, and any data experiencing more than a delay of 2τ is thrown away (i.e., no longer considered for scheduling), then the above rate guarantee translates to a delay guarantee of the form

$$\Pr(D^i \leq 2\tau) \geq 1 - \delta,$$

where D^i is the scheduling delay associated with a typical bit of data designated for user i.

To guarantee the required QoS, we will use a stochastic envelope based approach [5][9]. The idea is to lower bound the actual service $S_i(\tau)$ by a quantity $S_i^{low}(\tau)$ that satisfies two properties, firstly

$$S_i(\tau) \geq^{st} S_i^{low}(\tau),$$

so that if $S_i^{low}(\tau)$ meets the QoS guarantee then so will $S_i(\tau)$. Secondly, $S_i^{low}(\tau)$ will be analytically tractable from a resource allocation perspective.

We will first focus on providing the same QoS guarantee to all the real-time users, and later generalize to multiple QoS needs in Section III.

B. Opportunistic Round Robin

Recall that our goal is to find a scheduling scheme and resource allocation strategy that exploits both intra and inter class opportunism to provide high throughput to all users while meeting real-time users' QoS requirements. Yet, let us first consider scheduling n real-time users. A simple way to serve them is to use a frame with n slots. In every slot, the users feedback their rate for that slot and the base station opportunistically serves the best user. Once a user has been served in a frame, he does not compete for service until the next frame. This ensures that each active real-time user gets served once every frame. This scheme is similar to that proposed in [8], however the objective there was not to provide QoS guarantees. One might call this 'opportunistic round robin' scheduling. Consider a saturated system, i.e., all users have infinite backlogs. Under conventional round robin a typical user in such a system would see a slot whose rate distribution is the same as $X^{(1)}$, i.e., no opportunistic gain. However, under the opportunistic round robin scheme, a user is equally likely to be served on any one of the slots of the frame. If it is served on the $(n-j+1)^{th}$ slot, it would have competed with $j-1$ other users and will see a rate distribution of $X^{(j)}$. This means that the rate distribution in a *typical* slot will be a mixture, i.e., with probability (w.p.) $\frac{1}{n}$ it will see the distribution of $X^{(n)}$, w.p. $\frac{1}{n}$ a distribution of $X^{(n-1)}$ and so on. We let the random variable Y have the rate distribution seen by such a user, then

$$Y = \begin{cases} X^{(n)} \text{ w.p. } 1/n \\ \quad \ldots \quad \text{w.p. } 1/n \\ X^{(1)} \text{ w.p. } 1/n. \end{cases} \tag{1}$$

Clearly $Y \geq^{st} X^{(1)}$, so our opportunistic round robin scheme will give improved data rate to users.

In present day systems, a time slot is of the order of milliseconds (1.67 msec for CDMA-HDR), while video and multimedia traffic require guarantees of around 100 kbps on a time scale of the order of hundreds of milliseconds. So, if the number of real-time users is not large, i.e., frame sizes are tens of milliseconds, there is a 'slack' in the QoS requirement that is not exploited by opportunistic round robin which in turn can lead to severe system throughput penalties– our simulations (not presented here) show this. This slack can be used to schedule best effort users and enhance opportunism. An alternative is to have a larger frame and give multiple slots to users. This brings us to our proposed scheduling scheme.

C. Proposed Scheduling Scheme

In our scheme the frame is as long as the time period on which the QoS guarantees need to be ensured, i.e., τ. Each real-time user is assigned k 'tokens', i.e., each real-time user will be served at most k slots within a frame. Note that nk can at most be equal to τ. We describe how to determine the value of k in the next subsection.

The proposed scheduling scheme combines a policy to decide which users will be active, i.e., the set $A(t)$ that contend for a slot, with a mechanism to select the user to serve during that slot. To avoid confusion, we henceforth refer to the latter as 'selection criterion' and denote it by a set-valued function $\phi(\cdot)$. In this section we use max rate selection criterion among

users, i.e.,

$$\phi(B(t)) := \arg \max_{j \in B(t)} x^j(t),$$

where $B(t)$ is a set of users at time slot t. Note since rate distributions are i.i.d., i.e., symmetric, this criterion is fair and maximizes system throughput.

We present the proposed scheduling scheme in terms of an algorithm that is implemented every frame. It starts at the first slot of the frame and ends at the last slot of the frame. The time slots within a frame are indexed as $t = 1, \ldots, \tau$, while the number of tokens remaining for user j is denoted by k_r^j. Recall that $A_r(t)$ is the set of active real-time users allowed by the proposed scheduling scheme to compete during slot t, in contrast $A_b(t)$ is the set of best effort users that have data backlogged during slot t and $A(t) = A_r(t) \cup A_b(t)$.

Algorithm for the proposed scheduling scheme

---

1) Initialize $t = 1$ and $A_r(1)$ to be the set of all admitted real-time users each with k tokens allocated to it, i.e., $\forall j \in A_r(1)$, $k_r^j = k$.
2) If $t > \tau$, i.e., end of frame is reached, then go to Step 12, else if $(\tau - t) = \sum_{j \in A_r(t)} k_r^j$, i.e., the number of remaining slots in the frame is equal to the total number of remaining tokens, then go to Step 8 else go to the next step.

Phase I

3) Based on the feedback from the users, choose user i such that $i \in \phi(A(t))$, with ties broken randomly.
4) If i is a best effort user, then serve him and go to Step 7, else go to the next step.
5) If i is a real-time user which is backlogged, then serve him, else if $A_b(t)$ is not empty serve a best effort user from $A_b(t)$. The best effort user can be selected using any criterion e.g. proportionally fair, max rate etc.
6) Update $k_r^i = k_r^i - 1$, and if $k_r^i = 0$, i.e., user i has used up its tokens, then update $A_r(t) = A_r(t) \setminus \{i\}$, i.e., remove user i from $A_r(t)$.
7) Increment $t = t + 1$ and define $A_r(t + 1) = A_r(t)$. Go to Step 2.

Phase II

8) Based on the feedback, choose user i such that $i \in \phi(A_r(t))$, with ties broken randomly. Note that we are now choosing only among real-time users.
9) If i is a backlogged real-time user, then serve him, else if $A_b(t)$ is not an empty set, then serve a best effort user from $A_b(t)$. Again, the best effort user can be selected using any criterion e.g. proportionally fair, max rate etc.
10) Update $k_r^i = k_r^i - 1$, if $k_r^i = 0$, i.e., user i has used all of his tokens, then update $A_r(t) = A_r(t) \setminus \{i\}$, i.e., remove user i from $A_r(t)$.
11) Increment $t = t + 1$, if $t > \tau$, then go to the next step, else define $A_r(t + 1) = A_r(t)$ and go to Step 8.
12) Proceed to the next frame.

We now give a brief description of the scheme using an example containing a number of best effort sessions and 2 real-time users. Each real-time user is assigned 3 tokens. Figure 2 shows a frame of size $\tau = 10$.

Whenever a real-time user is given a chance to be served, his token count decreases by 1 and when the token count becomes zero, he is no longer considered for service (Steps 6 and 10).

The scheduling scheme is divided into two phases. During the first phase (Steps 3 - 7), both active real-time and best effort users are allowed to compete for service. In each slot, the user with the maximum rate is identified and served, with ties broken randomly. The first phase continues until the *total number of remaining tokens in the system is equal to the number of slots remaining in the frame* (Step 2). In our example (Figure 2), the first phase lasts until Slot 7. In Slot 3 and 5, real-time User 1 supported the highest data rate and was served, similarly real-time User 2 was served during Slot 6. Best effort users were served in the rest of the slots (the shaded ones). After Slot 7, the above mentioned condition for the end of first phase is satisfied, so the second phase starts.

During the second phase (Steps 8 - 11), only the real-time users are allowed to contend for service under the max rate selection criterion. This phase is needed to ensure that every real-time user is served as many times as the number of tokens assigned to it. Note that $A_r(t)$ will be empty by the end of the frame. The second phase slot assignment for the example are shown in Figure 2. Slots 8 and 9 are assigned to real-time User 2 and he is no longer allowed to compete. As a result, User 1 gets Slot 10, thus ensuring that both users got served as many times as the number of tokens they were allocated. Note that Figure 2 is just one of the many realizations that the proposed scheduling scheme could follow (even the starting point of the second phase is not fixed). In fact the number of possible realizations grows combinatorially in both n and k, making the scheme hard to analyze.

Note that we provision tokens for real-time users based on their QoS requirements, therefore it is possible that a real-time user selected for service may have no data to receive during that slot (note that the definition of an active real-time user allows this). When this is the case, we allow the slot to be used by any active best effort user (Steps 5 and 9). The best effort user to be served can be selected on a desirable criterion e.g. max rate, proportional fairness.

Some comments on the two phases of the proposed scheduling scheme. The first phase allows the exploitation of both inter and intra class opportunism and thus takes advantage of the slack in the QoS requirement. The second phase is needed to guarantee quality of service to real-time users, however note that opportunism is still exploited across the remaining real-time users. Together the two phases allow one to maintain high throughput while providing quality of service.

D. Analysis and Resource Allocation

The value of k must be decided so that the specified QoS guarantee is met for all real-time users. Let Z_j^* denote the data sent to a real-time user upon consuming its j^{th} token, i.e., the j^{th} time it gets served. Our goal is to determine the minimum

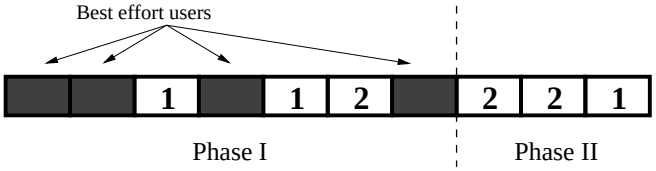


Fig. 2. Example of the proposed scheduling scheme with frame size of 10 and 2 real-time users each having 3 tokens

number of tokens k such that $\Pr(\sum_{j=1}^{k} Z_j^* \geq r\tau) \geq 1-\delta$. This is not easy to compute, even when users have i.i.d. rate distributions.

Note that $\forall j$, $Z_j^* \geq^{st} X$, i.e., at worst a user contends with no other users and thus sees the marginal channel capacity distribution of a typical slot with no opportunistic gain. Therefore it is likely that $\forall s$, $\sum_{j=1}^{s} Z_j^* \geq^{st} \sum_{j=1}^{s} X_j$, where X_j's are i.i.d. and $X_j \sim X$. (Note that because Z_j^*'s are not independent random variables $Z_j^* \geq^{st} X$ is not a sufficient condition for proving $\sum_{j=1}^{s} Z_j^* \geq^{st} \sum_{j=1}^{s} X_j$, but the bound will be shown to be true later.) Then perhaps, the simplest bound would be to replace Z_j^* by X, and finding the minimum value of k that satisfies $\Pr(\sum_{j=1}^{k} X_j \geq r\tau) \geq 1-\delta$ using e.g., the Central Limit Theorem. But this bound is very conservative, i.e., will allocate too many tokens, because X does not reflect any of the opportunistic gains achieved by the proposed scheduling scheme.

To find a more efficient, yet conservative resource allocation approach, consider a 'static division scheduling scheme', where the frame is divided into two parts. During the first part, consisting of $\tau - nk$ slots, the slots are opportunistically allocated among the best effort users, while the real-time users are opportunistically served during the second part. This is a special case of the original proposed scheduling scheme where only best effort users are served in Phase I. Let Z_j be the same quantity for the static division scheme as Z_j^* is for the proposed scheme. We claim in Theorem 2.2 that $\sum_{j=1}^{k} Z_j^* \geq^{st} \sum_{j=1}^{k} Z_j$, i.e., the static division scheduling scheme under performs relative to our proposed mechanism.

Before proving this claim, we digress to state three properties satisfied by both the proposed and the static division scheduling scheme when max rate selection is used under Assumption 2.1. These properties are used in the proof of Theorem 2.2, its supporting lemmas, and subsequent results.

Property 2.1: (*Equal Resource Allocation*) All real-time users are allocated an equal number k of tokens.

Property 2.2: (*Symmetrical Selection*) In a typical slot, each active real-time user is equally likely to be selected for service by the selection criterion (the selection probability for an active best effort user can be different).

Property 2.3: (*Monotonicity*) The selection criterion is such that for any user i and for any value of l, $X^{i,(l+1)} \geq^{st} X^{i,(l)}$, where $X^{i,(l)}$ is the random variable denoting the rate seen by user i given it is selected for service while competing with $l-1$ other users.

We now introduce some further notation. Let N_j^* be a random variable representing the number of real-time users in the system when a typical real-time user gets the j^{th} token in our proposed scheduling scheme. Let N_j represent the same quantity for the static division scheduling scheme. Note that under the static division scheduling scheme, a real-time user competes only with other real-time users while under the proposed scheduling scheme there might also be competing best effort users. Therefore it is likely that $N_j^* \geq^{st} N_j$, this is at the root of our next theorem, which is proven in Appendix I.

Theorem 2.2: Consider the proposed and the static division scheduling schemes where all real-time users are allocated an equal number k of tokens. Then under Assumption 2.1 and the max rate selection criterion, for a typical real-time user

$$\sum_{j=1}^{k} Z_j^* \geq^{st} \sum_{j=1}^{k} Z_j.$$

Theorem 2.2 implies that to meet the quality of service constraint, it is sufficient to satisfy $\Pr(\sum_{j=1}^{k} Z_j \geq r\tau) \geq 1-\delta$. To compute this, let us study the properties of Z_j. Note that Z_j is the maximum over N_j i.i.d. random variables with the same distribution as X. In other words, $Z_j \sim X^{(l)}$ $w.p.$ $\Pr(N_j = l)$, $\forall l$. The distribution of X is assumed to be known, but it is difficult to calculate the distribution of N_j because of the number of ways our opportunistic scheduling scheme can proceed, i.e., how users are served, grows in a combinatorial fashion. Also note that Z_j's are not i.i.d. which makes it difficult to calculate $\Pr(\sum_{j=1}^{s} Z_j \geq r\tau)$ for any given value of s. To remedy this, we propose a further stochastic lower bound that still factors the opportunistic gain. Our next claim is that $\sum_{j=1}^{k} Z_j \geq^{st} \sum_{j=1}^{k} Y_j$, where Y_j's are i.i.d. and $Y_j \sim Y$, where Y is as defined in (1) in Section II-B. We shall refer to this stochastic lower bound as the 'mixture bound'. The following theorem formally states our claim with the proof given in Appendix II.

Theorem 2.3: Consider the static division scheduling scheme where all real-time users are allocated an equal number of k tokens. Then under Assumption 2.1 and max rate selection criterion, for a typical real-time user

$$\sum_{j=1}^{k} Z_j \geq^{st} \sum_{j=1}^{k} Y_j,$$

where Y_j's are i.i.d. and $Y_j \sim Y$, with Y is as defined in (1).

Theorem 2.3 bounds the cumulative data received by a typical real-time user in a frame by a sum of i.i.d. random variables where each is a mixture of distributions. If the number of tokens required per user, i.e., k, is large enough, the distribution of $\sum_{j=1}^{k} Y_j$ can be roughly approximated, e.g. using the Central Limit Theorem. An advantage of using the Central Limit Theorem is that one can compute the value of k based only on the mean and variance of Y, which eliminates the need to know the actual distribution of X. Of course note that if users' rate distribution change, then so will the number of tokens required by them and the value of k will have to be recomputed and allocated to track such changes.

Note that by virtue of the definition of Y, the above approach factors the opportunistic gains in our scheme. Recall

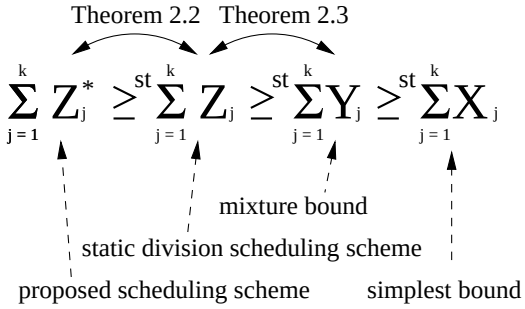


Fig. 3. Stochastic ordering of the cumulative data received by a typical real-time user in a frame under the proposed scheduling scheme, the static division scheduling scheme, the mixture method and the simplest method.

that the simplest bound to compute k conservatively would be to ensure $\Pr(\sum_{j=1}^{k} X_j \geq r\tau) \geq 1-\delta$. Now as discussed in Subsection II-B $Y \geq^{st} X$, and due to independence among Y_j's, $\sum_{j=1}^{k} Y_j \geq^{st} \sum_{j=1}^{k} X_j$. Hence once can conclude that $\sum_{j=1}^{k} Z_j^* \geq^{st} \sum_{j=1}^{k} X_j$, i.e., the simplest method is conservative. Figure 3 summarizes the overall stochastic ordering for the cumulative data received by a typical real-time user under the proposed scheduling scheme, the static division scheduling scheme, the mixture bound and the simplest bound.

A simple numerical experiment. As mentioned earlier, the simplest bound may allocate too many tokens. For example, we computed the number of required tokens per user using both the simplest bound and the proposed mixture bound for a system where each real-time user required a rate guarantee of 100 kbps over a time scale of 167 msec with an outage of 1%. The number of real-time users was 5 and all users were experiencing Rayleigh fading with a mean signal to noise ratio(SNR) of 2. Each slot was of size 1.67 msec (so the frame size was 100 slots) and the mapping from SNR to discrete rates was that used for CDMA-HDR [3]. The simple bound gave a requirement of 20 tokens per user while the mixture bound suggested only 12 tokens were needed. In addition, simulations showed that the exact number of tokens required to meet the guarantee were 11. *This suggests that our mixture bound is fairly tight, and thus useful*.

We emphasize that under the proposed scheme, unlike the weight based schemes discussed in related work, we were able to develop a concrete resource allocation approach.

III. Scheduling and Resource Allocation for Asymmetrical Channel Capacity Distributions

The symmetrical rate distributions case considered above, though unrealistic is a good starting point to solving the more general problem. In this section we allow users to experience different channel capacity distributions and describe the modifications required to our proposed scheme. We restate our assumption on the users' channel characteristics as follows:

Assumption 3.1: We assume the channel capacity (rate) for each user is a stationary ergodic process and these processes are independent, but *not* necessarily identically distributed across users. The channel capacity for each user is fast fading, i.e., the channel capacity for each user is independent across slots and remains constant during a slot. Further we assume that the marginal distribution for each user is either known a priori, or estimated by the base station.

Note again that we are not assuming any specific distribution on the channel capacity variation.

The token scheme proposed in Section II achieves multiple goals, it guarantees that the QoS requirements for real-time users are met, while exploiting both intra and inter class opportunism to achieve high overall throughput. We want these desirable properties to hold while extending the scheme to the asymmetric case. Our approach of allocating tokens to each real-time user and then scheduling users opportunistically allows us to achieve these goals. However to efficiently calculate the number of tokens required by a user, one would like the Theorem 2.2 and Theorem 2.3 to also hold under Assumption 3.1. As mentioned earlier, the proofs of these theorems depend on the Properties 2.1, 2.2 and 2.3 holding true.

Let us consider the 'Equal Resource Allocation' property. Under Assumption 3.1, it is likely that different users may require different number of tokens to be guaranteed the same QoS. This can be dealt with by simply over allocating tokens so that all real-time users have the same number of tokens, but this in turn can lead to lesser number of real-time users getting admitted. Better alternatives will be discussed later.

The 'Symmetrical Selection' and 'Monotonicity' properties depend on the selection criterion. It is clear that under Assumption 3.1, it is unlikely that max rate selection criterion will satisfy Property 2.2. An alternative is to randomly select a user (among the active ones), however there would be no opportunistic gains in this case. Our solution is to use maximum quantile scheduling, which will ensure that the two properties are satisfied and yet give good opportunistic gains. Maximum quantile scheduling has been proposed by several researchers under different guises [11][12][4][15], it is briefly introduced in the next subsection.

A. Maximum Quantile Scheduling

We introduce some notation to describe maximum quantile scheduling. The rate distribution function of the i^{th} user, i.e., the distribution function of X^i is denoted by $F_i(\cdot)$ and its unique inverse by $F_i^{-1}(\cdot)$. For simplicity, we consider X^i to be *continuous random variables*. The results can be extended to the discrete case [13].

As mentioned earlier, the idea of the scheme is to schedule a user whose current rate is highest relative to his *own* distribution, i.e., in the highest quantile. Under maximum quantile scheduling, user i is selected for service on time slot t when [12]

$$i \in \arg\max_{j \in A(t)} F_j(x^j(t)).$$

Using the fact that $F_j(X^j)$ is uniformly distributed on $[0,1]$, one can show that each competing user is *equally likely* to get served on a typical slot, i.e., Symmetrical Selection is satisfied by the scheme.

Next we show that maximum quantile scheduling satisfies Property 2.3, i.e., Monotonicity. Define $X^{i,(l)} = \max\{X_1^i, \ldots, X_l^i\}$, where X_j^i's are i.i.d. and $X_j^i \sim X^i$. Then

the rate experienced by user i when selected for service on a typical slot by maximum quantile scheduling while competing with $l-1$ other users, has the same distribution as $X^{i,(l)}$. It is easy to see that for any l, $X^{i,(l+1)} \geq^{st} X^{i,(l)}$, i.e., Monotonicity is satisfied.

Note that maximum quantile scheduling requires that users' rate distributions be known at the access point. However, rate distributions can be estimated using the feedback sent by users' on each slot. Let R^i denote the rate seen by user i on a typical slot in which it is selected for service under maximum quantile scheduling with perfectly known distributions. Let $\tilde{R}^i_m$ denote the same quantity for user i under maximum quantile scheduling with users' rate distribution being estimated using m previous samples of the feedback. Then it is established in [14] that $\forall r$,

$$(\frac{m+1}{n_{tot}}(1-(\frac{m}{m+1})^{n_{tot}})) \leq \frac{\Pr(\tilde{R}^i_m \leq r)}{\Pr(R^i \leq r)} \leq 1.$$

Recall that here n_{tot} denotes the total number of users in the system. The above statement can be simplified to show that if one needs to achieve an average throughput penalty of less than ϵ due to rate distribution estimation error, then one needs only $m = \frac{n_{tot}}{2\epsilon}$ samples, i.e., linear with the number of users. For example, to achieve a penalty less than 5%, $10n_{tot}$ samples are needed, which seems reasonable since there are hundreds of slots in a second.

We are now in a position to describe the proposed modification to our scheduling discipline under Assumption 3.1.

B. Proposed Modification

We begin by discussing resource allocation, i.e., evaluating how many tokens should be allocated to each user. In order to do so, we define a new quantity Y^i given by

$$Y^i = \begin{cases} X^{i,(n)} & \text{w.p. } 1/n \\ \dots & \text{w.p. } 1/n \\ X^{i,(1)} & \text{w.p. } 1/n. \end{cases} \quad (2)$$

As mentioned earlier, it is likely that due to the asymmetric nature of users rate distributions, each real-time user may require a different number of tokens for the same QoS requirement. For each real-time user, calculate

$$k^i = \min_s \{s \mid \Pr(\sum_{j=1}^{s} Y^i_j \geq r\tau) \geq 1-\delta\}, \quad (3)$$

where Y^i_j are i.i.d. and $Y^i_j \sim Y^i$, with Y^i defined in (2). We shall let k now be given by

$$k = \max_{j=1,\dots,n} k^j. \quad (4)$$

Suppose every real-time user is allocated k tokens. Note that we require that, $nk \leq \tau$, i.e., the total number of tokens allocated must be less than or equal to the size of frame.

It should be clear by now that the selection criterion is changed to maximum quantile instead of max rate. Thus the selection criterion in the algorithm is now defined as

$$\phi(B(t)) := \arg \max_{j \in B(t)} F_j(x^j(t)),$$

when the X^j are continuous. For the discrete case, we refer the reader to [14][4].

With the two proposed modifications, the three properties stated in the previous section are satisfied. It follows that the claims of Theorem 2.2 and 2.3 hold under Assumption 3.1. This in turn shows that the value of k obtained in (3) and (4) will be conservative.

Rather than state the modified versions of Theorem 2.2 and 2.3 under Assumption 3.1, we will state a stronger version that will be useful later in the sequel. Let S be any set such that $S \subseteq \{1,\dots,k\}$, this can be viewed as any subset of the tokens assigned to a user. Let Z^{i*}_j denote the transmitted data to real-time user i when it uses up the j^{th} token under the proposed scheduling scheme and let Z^i_j be the same quantity for the static division scheduling scheme. The following are the generalized theorem statements without proofs (which are analogous to those of Theorem 2.2 and 2.3).

Theorem 3.2: Consider the proposed and the static division scheduling schemes where all real-time are allocated an equal number k of tokens. Then under Assumption 3.1 and maximum quantile selection criterion, for any real-time user i

$$\sum_{j \in S} Z^{i*}_j \geq^{st} \sum_{j \in S} Z^i_j,$$

for $S \subseteq \{1,\dots,k\}$.

Theorem 3.3: Consider the static division scheduling scheme where all real-time users are allocated an equal number k of tokens. Then under Assumption 3.1 and maximum quantile selection criterion, for any real-time user i

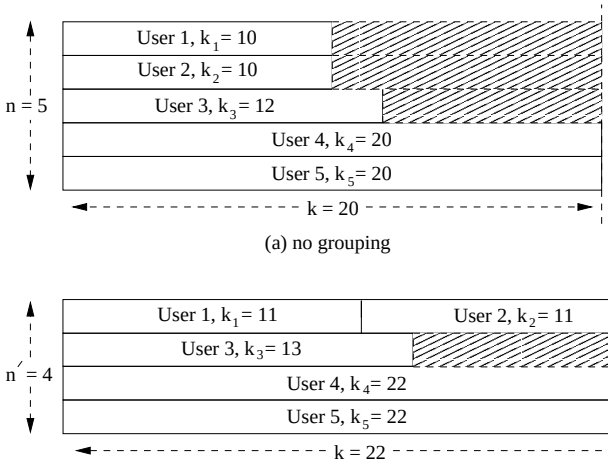
$$\sum_{j \in S} Z^i_j \geq^{st} \sum_{j \in S} Y^i_j,$$

for $S \subseteq \{1,\dots,k\}$, where Y^i_j's are i.i.d. with the same distribution as Y^i is given by (2).

Grouping of users. As mentioned earlier, allocating the same number of tokens k is based on (3) and (4) is likely to be conservative for heterogeneous users. To improve upon this, we group users with smaller token requirements into single virtual users. We explain this with an example below.

Consider the following scenario, suppose there are 5 real-time users in the system. All users undergo Rayleigh fading, but have different mean SNR. User 1 and 2 have a mean SNR of 3, User 3 has a mean SNR of 2, while User 4 and 5 have a mean SNR of 0.8. The SNR to rate mapping is same as the example discussed in Section II-D, i.e., same as that of CDMA-HDR. All real-time users are to meet a QoS guarantee of 100 kbps over a time scale of 167 msec with an outage probability of 1%. The frame size is thus 100 slots.

If tokens are allocated according to (3) and (4), then each real-time user would be allocated 20 tokens each (see Figure 4(a)), and there would be no slots left for Phase I of the proposed scheduling scheme. However, if a given real-time user competes with at most 3 other real-time users in a slot, then $Y^i = X^{i,(l)}$ $w.p.$ $\frac{1}{4}$, $l = 1,\dots,4$. In this case Users 1 and 2 will require 11 tokens each, while User 3 requires 13 tokens and User 4 and 5 require 22 tokens each. One can then



(a) no grouping

(b) User 1 and 2 grouped together

Fig. 4. Parts (a) and (b) show token requirements and allocation with and without grouping respectively. The shaded portion depicts the excess allocated tokens

allocate 22 tokens to User 3, 4 and 5 and combine User 1 and 2 into a single virtual user having a total of 22 tokens. This is illustrated in Figure 4(b), there n' represents the maximum number of real-time users that are allowed to compete for a slot (note that $n' = n$ if no grouping is used, else $n' < n$). As shown, User 1 uses the first 11 tokens of the virtual user followed by User 2. Then $S = \{1, \ldots, 11\}$ for User 1 and by Theorem 3.2 and 3.3, both would be able to meet their QoS requirement. We simulated such a system with 16 best effort users and verified our claim to be true. The advantage of grouping is exhibited in this example where instead of 100 tokens, only 88 need to be allocated to real-time users.

There are multiple ways of grouping users together, one can also group more than two users. Another possibility is to increase k slightly to allow better groupings. Referring back again to our example, suppose User 3 and 4 required 21 tokens each instead of the 22 required (with grouping), then one could have defined k as 22, i.e., over allocate by 1 token to User 3 and 4, to allow us to group Users 1 and 2.

Unfortunately finding the optimal grouping is an NP-Hard problem. We introduce some notation to prove our claim. Let $\mathcal{P}_{n'}$ denote the collection of all partitions of the set of all real-time users with n' non-empty sets. Let P denote a partition of the set of all real-time users, and p be a set in P. We denote the tokens required by user i when it is competing for service with n' virtual real-time users for service by $k^i(n')$. Then the problem of optimal grouping can be written as follows:

Optimal grouping problem: Find the number of groups n' and a partition P of all real-time users into that number of groups such that

$$\min_{n'=1,\ldots,n} n' k_{max}(n'),$$

where

$$k_{max}(n') = \min_{P \in \mathcal{P}} \max_{p \in P} \sum_{i \in p} k^i(n').$$

The following theorem shows that the above defined problem of optimal grouping is NP-Hard.

Theorem 3.4: The Optimal grouping problem is NP-Hard.

Proof: Consider a fixed n', then finding the value of $k_{max}(n')$ is equivalent to the load balancing problem, which in turn is known to be NP-Hard [6]. ■

One can however propose simple heuristics to find suboptimal grouping solutions. For example consider a given n', then a user must belong to one of the n' groups, each corresponding to a single/virtual user. A simple solution would be to order users by their 'load' $k^i(n')$ and starting with the highest $k^i(n')$, place them in a group that currently has the lowest total load. One can search over different best fit solutions varying values of n' and find the best solution. For other heuristics, see [6].

Multiple QoS Guarantees. Let us consider providing different rate guarantees to different users. Here, each user can ask for a specific rate guarantee r^i with his own outage probability δ^i. However, the time scale over which the guarantee is given, i.e., the frame length τ is common to all users. (One can somewhat relax this constraint by giving guarantees over integral multiples of τ.) Supporting multiple QoS requirements can lead to different users needing different numbers of tokens, which can be solved by grouping real-time users together. Thus extending our scheme to meet multiple QoS criteria efficiently.

C. Call Admission Policy

The call admission policy is quite simple, to admit a call $n'k \leq \tau$, where k now is the number of tokens allocated to each user or a virtual user (if there is grouping).

However, note that in order to check whether a new user can be admitted into the system we have assumed that the capacity distribution of the new user $F_{n+1}(\cdot)$ is known a priori, this is unlikely. A practical solution to this problem is to initially use a typical distribution derived from users currently or previously associated with the wireless access point. For example, let $\tilde{F}(\cdot)$ be the ensemble average of the distributions for ongoing (or past) users, e.g., $\tilde{F}(x) = \frac{\sum_{j=1}^{n_{tot}} F_j(x)}{n_{tot}}$. This distribution represents what a typical user might see, or what a mobile user might see throughout its lifetime in the system.

It is also important to note here that call admission is a long term decision, and one may need to save resources for future events like time varying rate distributions. Here, the number of tokens required by a user may vary across frames, this can be due to inaccuracy in estimating the distribution of users (especially for the newly admitted user) and time varying nature of users' rate distributions. Therefore one needs to reserve a pool of extra slots to handle such variations and allocate tokens from the pool to users that are not able to meet their QoS requirement in a frame. This pool can also be used for incoming handoffs from neighboring cells. Estimating the number of tokens can be investigated as future work.

IV. Simulation Results

We simulated the proposed scheme under various scenarios. We begin by considering the performance of the scheme as the number of real-time users and the QoS constraint vary. Next we observe the outage of real-time users with an

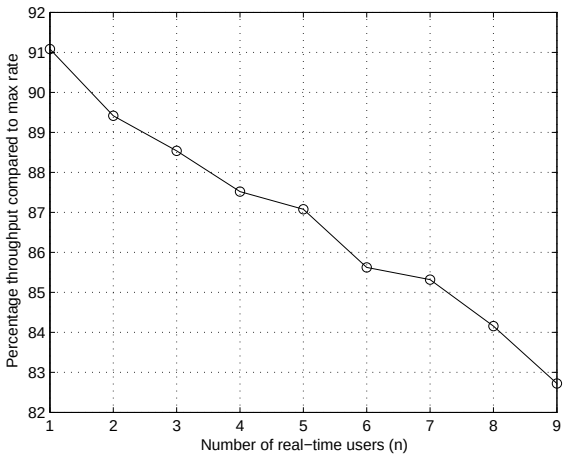


Fig. 5. Percentage system throughput achieved by the proposed scheduling scheme compared to max rate scheduling with increasing number of real-time users.

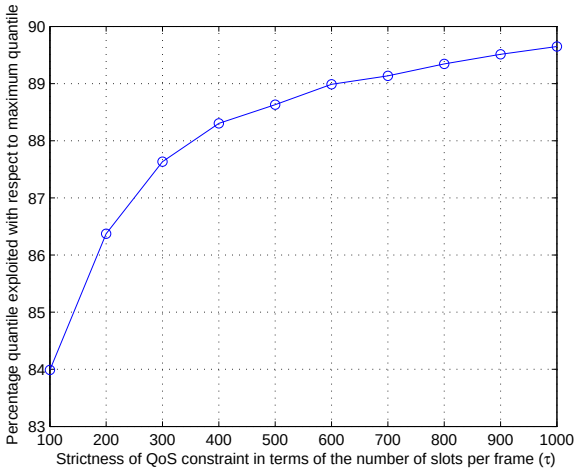


Fig. 6. Percentage average quantile achieved by the proposed scheduling scheme compared to maximum quantile scheduling with varying QoS constraint in number of slots per frame.

increasing number of best effort users. Finally we propose a heuristic to accommodate slow fading channels and observe its performance. Our simulation setup is similar to that of CDMA-HDR, the slot time period was set to 1.67 ms, with SNR to rate mapping borrowed from [3].

A. Throughput & Opportunism Performance

In the first simulation, we investigate the overall system throughput as the number of real-time users increases. For a reasonable comparison of the throughput performance, in this simulation all users have i.i.d. Rayleigh fading channel capacity distributions with a mean SNR of 2. Each real-time user requires a guarantee of 100 kbps over a time scale of 167 msec (100 slots) with an outage of 1%. The total number of users is fixed at 20, while the number of real-time users increases from 1 to 9. For a given number of real-time users, the number of tokens required by each user was calculated using the mixture bound and the system was simulated by allocating these resources to each real-time user. To put our throughput results in perspective, we theoretically calculated the overall system throughput that would be achieved by the 20 users under max rate scheduling with no QoS constraint. Here, we remind the reader that maximum rate scheduling *maximizes overall system throughput* that can be achieved. The throughput achieved by our scheme as a percentage relative to this upper theoretical bound is are plotted in Figure 5. The first observation is that we are able to achieve more than 90% of throughput with 1 real-time user. Second, note that while the number of real-time users increases from 1 to 9, the throughput degradation experienced is less than 9%. This indicates that our scheme is quite robust to increases in the number of real-time users in terms of degradation in the overall system throughput.

In our second set of simulations, we studied the tradeoff in the overall system performance as the QoS requirements were relaxed. Here we allowed the users to undergo heterogeneous fading. The setup is the same as the one used in describing grouping in Section III. There are 5 real-time user all undergoing Rayleigh fading with mean 3, 3, 2, 0.8 and 0.8, along with 16 best effort users, also experiencing Rayleigh fading with a mean SNR of 2. In our simulations, we grouped the first two real-time users (as discussed in the grouping example). Each real-time user was given a guarantee of 100 kbps with an outage of 1% over varying frame sizes. The number of slots in a frame was varied from 100 slots to 1000 slots in steps of 100 slots.

In the heterogeneous case, comparing the performance of our scheme to max rate scheduling is not reasonable. Therefore, in this simulation we kept track of the average quantile of the user served by our scheme, i.e., $E[\sum_{i=1}^{n_{tot}} F_i(X^i)\mathbf{1}_{S^i}]$, where $\mathbf{1}_{S^i}$ is the indicator function of the event S^i, which is the event that user i gets served under our scheme. Note that opportunistic scheduling tries to serve the user that is currently experiencing a 'good' rate. A measure of the goodness of the current rate can be the quantile of the current rate of the user, i.e., $F_i(x^i(t))$ [14]. Therefore the average quantile of the user served under a scheme is a measure of opportunism being exploited by the scheme. To again put our results in perspective we plotted our results as a percentage of the average quantile of the user served under maximum quantile scheduling without any QoS constraints). Note that by definition maximum quantile scheduling will maximize the quantile of the user being served.

The results are plotted in Figure 6. We note that even for strictest constraint a large part of opportunism, i.e., 84% is exploited, and this grows to almost 90% as the constraint loosens.

B. Outage versus Number of Best Effort Users

We now study the outage experienced by a real-time user as the number of best effort users increases. This is interesting because as the number of best effort users increase, real-time users are more likely to get served only during the second phase of the proposed scheduling scheme. Since the second phase is less throughput efficient than the first, the outage probability of a real-time user should increase with the number of best effort users. However since our bounds are calculated

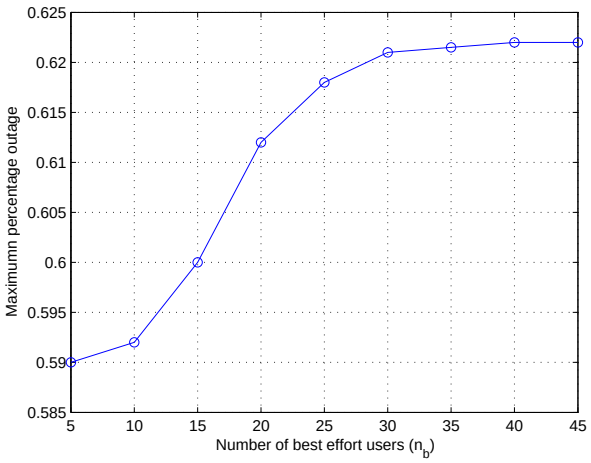


Fig. 7. Maximum percentage outage experienced by real-time users with increasing number of best effort users.

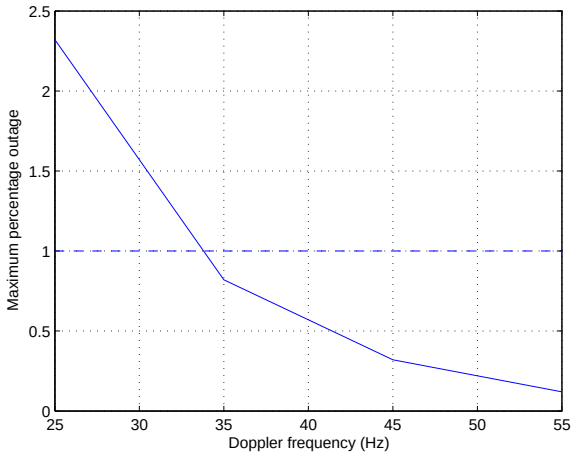


Fig. 8. Mean percentage outage experienced by real-time users under token borrowing scheme and slow fading channels. The dotted line represents the target outage

for the worst case scenario, each real-time user should still be able to meet his requirement, this is verified by our simulation results.

The setup was similar to the heterogeneous case in the previous subsection. Each real-time user required a guarantee of 100 kbps over a time scale of 167 msec (100 slots) with an outage of 1%. While the number of real-time users was 5, the number of best effort users was varied from 5 to 45 in steps of 5. Figure 7 shows the results in terms of the maximum percentage outage experienced among all the real-time users. Observe that the maximum outage is way below the target outage of 1%. We also note that the outage probability flattens out as the number of best effort users increase, this is because real-time users are now mostly served in the second phase of the scheme.

C. Outage Performance under Slow Fading

Our analysis in this paper assumes fast fading channels, this may be optimistic. Our simulations suggested that the proposed scheme does not perform well under slow fading channel conditions. We remedy this by proposing a heuristic.

In a frame some real-time users may be experiencing higher fades than their mean, while other real-time users might be suffering low fades. Then those experiencing high fades will require fewer allocated tokens and vice versa. This immediately suggests the possibility of token borrowing among users, i.e., users undergoing high fade allow other users to borrow some of their slots. If a real-time user satisfies his data requirement before finishing his allocated tokens, his remaining tokens are placed in a virtual pool. Whenever a real-time user finishes his allocated quota of tokens without satisfying his requirement, he can borrow tokens from the virtual pool until his requirement is satisfied or the pool is exhausted.

We simulated the performance of the proposed heuristic under varying degree of correlation (in time) of users' channel. There were a total of 20 users in the system with 5 real-time users with heterogeneous channel capacities as described in the previous subsection. Each real-time user is given a guarantee of 100 kbps over a time scale of 167 msec with an outage of 1%. The degree of correlation in a user's channel is varied using Doppler frequency from 25 Hz to 55 Hz in steps of 10 Hz. The maximum percentage outage experienced across real-time users observed for each step are plotted in Figure 8. Observe that the proposed heuristic is able to meet its requirement for Doppler frequencies higher than or equal to 35 Hz. Note that when the scheme is unable to meet the QoS criterion, one can suitably modify the size of the pool using history (so as to meet the guarantee).

V. Conclusion

In this paper we proposed a scheduling and resource allocation scheme that allowed base station to serve a mixture of real-time and best effort users. The proposed scheme realizes probabilistic QoS guarantees over short time scales to real-time users while exploiting both intra and inter class opportunism across users. The effectiveness of the proposed approach is validated by simulation results. The proposed scheme also did away with the conventional approach of providing QoS by tuning relative weights among users. We also developed a simple call admission policy for the proposed scheme. A unique advantage of the proposed approach is that it supports users with arbitrary channel capacity distributions, this makes the scheme amenable to real world scenarios. Finally we proposed a heuristic for channels with slow fading characteristics.

Appendix I
Proof of Theorem 2.2

Before presenting the proof, we introduce some notation. A vector of quantities say W_j is represented as $\overrightarrow{W}_{1:l} = (W_1, \ldots, W_l)$. For any two vectors $\overrightarrow{W}_{1:l}$, $\overrightarrow{V}_{1:l}$, $\overrightarrow{W}_{1:l} \geq \overrightarrow{V}_{1:l}$ means that for all $j = 1, \ldots, l$, $W_j \geq V_j$. In other words, $\overrightarrow{W}_{1:l}$ is componentwise greater than $\overrightarrow{V}_{1:l}$. Recall that N_j^* is the random variable representing the number of active real-time users in the system when a typical real-time user gets the j^{th} token in the proposed scheduling scheme and N_j be the same quantity for the static division scheduling scheme.

Then $\overrightarrow{N}^*_{1:k}$ and $\overrightarrow{N}_{1:k}$ are the vector representation of N^*_j and N_j respectively. We begin by proving the following lemma.

Lemma 1.1: Consider the proposed and static division scheduling scheme with all real-time users being allocated an equal number of k tokens. Then for a typical real-time user under Assumption 2.1 and max rate selection criterion

$$\Pr(\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) = \Pr(\overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}) \tag{5}$$

for any vector $\overrightarrow{n}_{1:k}$.

Proof: For the proposed scheduling scheme, consider only those slots in which real-time users are served. There are exactly nk slots of this type. If one considers the relative slot assignment possibilities among real-time users in these nk slots, then the number of possible realizations is $\binom{nk}{k\ldots k}$.

Now consider a slot among these nk slots, say the l^{th} one. Then due to Property 2.2, every active real-time user during that slot is equally likely to get selected for service. Again due to Property 2.1 and 2.2, every real-time user is equally likely to be competing or active during the l^{th} slot. Then if we average over all realizations of the proposed scheduling scheme, each user is equally likely to get assigned the l^{th} slot.

Then the probability of a realization of the proposed scheduling scheme in terms of the assignment of the nk slots among the real-time users is given by $\frac{1}{\binom{nk}{k\ldots k}}$. Consider such realizations with $\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}$ for a particular real-time user. Let there be $h_{\overrightarrow{n}_{1:k}}$ be such realizations, i.e, with $\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}$ for the user. Then

$$\Pr(\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) = \frac{h_{\overrightarrow{n}_{1:k}}}{\binom{nk}{k\ldots k}}.$$

Similarly for the static division scheduling scheme,

$$\Pr(\overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}) = \frac{h_{\overrightarrow{n}_{1:k}}}{\binom{nk}{k\ldots k}}.$$

Then clearly

$$\Pr(\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) = \Pr(\overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}).$$

■

Next we present the proof for Theorem 2.2.

Proof: Recall that $\sum_{j=1}^k Z^*_j \geq^{st} \sum_{j=1}^k Z_j$ means that for any z,

$$\Pr(\sum_{j=1}^k Z^*_j \geq z) \geq \Pr(\sum_{j=1}^k Z_j \geq z). \tag{6}$$

To prove this, we will show that for any vector $\overrightarrow{z}_{1:k} = (z_1, \ldots, z_k)$,

$$\Pr(\overrightarrow{Z}^*_{1:k} \geq \overrightarrow{z}_{1:k}) \geq \Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k}).$$

Conditioning on the number of real-time users present in each of the slots, we get

$$\sum_{\overrightarrow{n}_{1:k}} \Pr(\overrightarrow{Z}^*_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) \Pr(\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) \geq \sum_{\overrightarrow{n}_{1:k}} \Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}) \Pr(\overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}). \tag{7}$$

From Lemma 1.1, we know that

$$\Pr(\overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) = \Pr(\overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}).$$

Then to prove (7), we need to show that

$$\Pr(\overrightarrow{Z}^*_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}) \geq \Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}). \tag{8}$$

Let M^*_j be the random variable representing the number of active best effort users when a typical real-time user gets selected the j^{th} time. Then $\overrightarrow{M}^*_{1:k}$ is the vector representation of the M^*_j. Conditioning left hand side of (8) on M^*_j, we get

$$\sum_{\overrightarrow{m}_{1:k}} \Pr(\overrightarrow{Z}^*_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}, \overrightarrow{M}^*_{1:k} = \overrightarrow{m}_{1:k}) \Pr(\overrightarrow{M}^*_{1:k} = \overrightarrow{m}_{1:k}) \geq \Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}).$$

We also know that channel variations are independent across slots, thus we have that

$$\Pr(\overrightarrow{Z}^*_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}^*_{1:k} = \overrightarrow{n}_{1:k}, \overrightarrow{M}^*_{1:k} = \overrightarrow{m}_{1:k}) = \Pr(Z^*_1 \geq z_1 | N^*_1 = n_1, M^*_1 = m_1) \ldots \Pr(Z^*_k \geq z_k | N^*_k = n_k, M^*_k = m_k),$$

and

$$\Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k} | \overrightarrow{N}_{1:k} = \overrightarrow{n}_{1:k}) = \Pr(Z_1 \geq z_1 | N_1 = n_1) \ldots \Pr(Z_k \geq z_k | N_k = n_k).$$

Therefore to prove (6), we need to prove that $\forall j$,

$$\Pr(Z^*_j \geq z_j | N^*_j = n_j, M^*_j = m_j) \geq \Pr(Z_j \geq z_j | N_j = n_j)$$

This is clearly true from Property 2.3. ■

APPENDIX II
PROOF OF THEOREM 2.3

We present a few lemmas before proving the theorem.

Lemma 2.1: Given any sequence of non-negative numbers a_l, b_l and c_l, $l = 1, \ldots, n$. If $\forall\ l, j,$ s.t. $l > j$, $a_l \geq a_j$ and $\forall h = 1, \ldots, n$, $\sum_{l=h}^n b_l \geq \sum_{l=h}^n c_l$, then $\sum_{l=1}^n a_l b_l \geq \sum_{l=1}^n a_l c_l$.

Proof: We know that $\forall h$, $\sum_{l=h}^n b_l \geq \sum_{l=h}^n c_l$ and $\forall\ l, j,$ st $l > j$, $a_l \geq a_j$. So $\forall h$,

$$(a_h - a_{h-1})(\sum_{l=h}^n b_l) \geq (a_h - a_{h-1})(\sum_{l=h}^n c_l),$$

where a_0 is defined to be equal to 0. Summing over all h, we get

$$\sum_{h=1}^n \{(a_h - a_{h-1})(\sum_{l=h}^n b_l)\} \geq \sum_{h=1}^n \{(a_h - a_{h-1})(\sum_{l=h}^n c_l)\}.$$

This simplifies to $\sum_{l=1}^n a_l b_l \geq \sum_{l=1}^n a_l c_l$.

■

Lemma 2.2: Consider the static division scheduling scheme with all real-time users being allocated an equal number of k tokens and max rate selection criterion. Then under

Assumption 2.1, the data received by a typical real-time user when it gets served for the last time, i.e., the k^{th} time has the same distribution as Y, i.e., $Z_k \sim Y$.

Proof: Due to Property 2.1 and 2.2, the probability that a user is the first one to leave the system, i.e., be selected for service k times is $1/n$. If it is the first user to leave the system, then $Z_k \sim X^{(n)}$. Similarly for any value of j, the probability that a user gets selected the k^{th} time when there are a total of j active real-time users in the system is $1/n$. Then for that user, $Z_k \sim X^{(j)}$. Hence, $Z_k \sim Y$. ∎

Define a set,

$$\begin{aligned} S_{\bar{n}_l} \quad = \quad & \{\overrightarrow{n}_{l+1:k} | \exists n_j \text{ in } \overrightarrow{n}_{l+1:k} \text{ s.t. } n_j \geq \bar{n}_l \text{ and} \\ & \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = \overrightarrow{n}_{l+1:k}) \geq \\ & \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = (\bar{n}_l, \ldots, \bar{n}_l))\}. \end{aligned}$$

Lemma 2.3: For any $\overrightarrow{n}_{l+1:k} \notin S_{\bar{n}_l}$,

$$\begin{aligned} & \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = \overrightarrow{n}_{l+1:k}) \leq \\ & \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = (\bar{n}_l, \ldots, \bar{n}_l)). \end{aligned}$$

Proof: We give the proof by contradiction. Assume that $\exists \overrightarrow{n}_{l+1:k} \notin S_{\bar{n}_l}$ s.t. $\Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = \overrightarrow{n}_{l+1:k}) \geq \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = (\bar{n}_l, \ldots, \bar{n}_l))$. Now given the number of users present in the system, the data transferred in a slot is independent of other slots. So,

$$\begin{aligned} & \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = \overrightarrow{n}_{l+1:k}) = \\ & \Pr(Z_{l+1} \geq z_{l+1} | N_{l+1} = n_{l+1}) \ldots \Pr(Z_k \geq z_k | N_k = n_k) \end{aligned}$$

and

$$\begin{aligned} & \Pr(\overrightarrow{Z}_{l+1:k} \geq \overrightarrow{z}_{l+1:k} | \overrightarrow{N}_{l+1:k} = (\bar{n}_l, \ldots, \bar{n}_l)) = \\ & \Pr(Z_{l+1} \geq z_{l+1} | N_{l+1} = \bar{n}_l) \ldots \Pr(Z_k \geq z_k | N_k = \bar{n}_l). \end{aligned}$$

Since $\overrightarrow{n}_{l+1:k} \notin S_{\bar{n}_l}$, then $\forall j$, $n_j < \bar{n}_l$. So $\forall j$, by Property 2.3

$$\Pr(Z_j \geq z_j | N_j = \bar{n}_l) \geq \Pr(Z_j \geq z_j | N_j = n_j).$$

This contradicts our assumption. ∎

We prove Theorem 2.3 now.

Proof: The goal is to show $\sum_{j=1}^{k} Z_j \geq^{st} \sum_{j=1}^{k} Y_j$, i.e., $\forall\ z$,

$$\Pr(\sum_{j=1}^{k} Z_j \geq z) \geq \Pr(\sum_{j=1}^{k} Y_j \geq z).$$

To prove this, we will show that for any vector $\overrightarrow{z}_{1:k} = (z_1, \ldots, z_k)$,

$$\Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k}) \geq \Pr(\overrightarrow{Y}_{1:k} \geq \overrightarrow{z}_{1:k}).$$

Since the Y_j's are independent, this simplifies the above inequality to

$$\Pr(\overrightarrow{Z}_{1:k} \geq \overrightarrow{z}_{1:k}) \geq \Pr(Y_1 \geq z_1) \ldots \Pr(Y_k \geq z_k). \quad (9)$$

Using conditioning, we can rewrite the left side of (9) as

$$\begin{aligned} & \Pr(Z_1 \geq z_1 | \overrightarrow{Z}_{2:k} \geq \overrightarrow{z}_{2:k}) \ldots \Pr(Z_j \geq z_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}) \\ & \qquad \ldots \Pr(Z_k \geq z_k) \end{aligned}$$

Then (9) can be proved if we show that $\forall j$,

$$\Pr(Z_j \geq z_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}) \geq \Pr(Y_j \geq z_j).$$

Conditioning on the number of users present in the system during the j^{th} time when the user is served,

$$\begin{aligned} & \textstyle\sum_{n_j=1}^{n} \{\Pr(Z_j \geq z_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}, N_j = n_j) \\ & \qquad \Pr(N_j = n_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k})\} \geq \\ & \textstyle\sum_{n_j=1}^{n} \{\Pr(Y_j \geq z_j | \tilde{N}_j = n_j) \Pr(\tilde{N}_j = n_j)\}, \end{aligned} \quad (10)$$

where all $\tilde{N}_j$ are i.i.d. and uniformly distributed on $\{1, \ldots, n\}$ (from (1)). Given the number of users in a slot, the data obtained is independent of data received in other slots, so

$$\begin{aligned} & \Pr(Z_j \geq z_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}, N_j = n_j) \\ & \qquad = \Pr(Z_j \geq z_j | N_j = n_j). \end{aligned}$$

Note that when $N_j = n_j$, then a user will have to compete among n_j users to get service in the slot, so

$$\Pr(Z_j \geq z_j | N_j = n_j) = \Pr(X^{(n_j)} \geq z_j).$$

Also from equation (1), we have

$$\Pr(Y_j \geq z_j | \tilde{N}_j = n_j) = \Pr(X^{(n_j)} \geq z_j).$$

We can simplify (10) to,

$$\begin{aligned} & \textstyle\sum_{n_j=1}^{n} \Pr(X^{(n_j)} \geq z_j) \Pr(N_j = n_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}) \geq \\ & \textstyle\sum_{n_j=1}^{n} \Pr(X^{(n_j)} \geq z_j) \Pr(\tilde{N}_j = n_j) \end{aligned} \quad (11)$$

From Lemma 2.1, (11) can be proved if $\forall l$,

$$\Pr(X^{(l+1)} \geq z_j) \geq Pr(X^{(l)} \geq z_j) \quad (12)$$

and $\forall \bar{n}_j$,

$$\Pr(N_j \geq \bar{n}_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}) \geq \Pr(\tilde{N}_j \geq \bar{n}_j) \quad (13)$$

From Property 2.3, it is clear that (12) is true. To prove (13), first consider the right hand side of the equation. Referring to Lemma 2.2, we get

$$\begin{aligned} & \Pr(\tilde{N}_j \geq \bar{n}_j) = \Pr(N_k \geq \bar{n}_j) = \\ & \textstyle\sum_{\overrightarrow{n}_{j+1:k} | n_k \geq \bar{n_j}} \Pr(\overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}). \end{aligned} \quad (14)$$

We know that $\forall j$, $N_j \geq N_{j+1}$ almost surely. Thus $\{\overrightarrow{n}_{j+1:k} | n_k \geq \bar{n_j}\} \subseteq S_{\bar{n}_j}$, then from (14) we get

$$\sum_{\overrightarrow{n}_{j+1:k} \in S_{\bar{n}_j}} \Pr(\overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}) \geq \Pr(\tilde{N}_j \geq \bar{n}_j). \quad (15)$$

Now consider the left hand side of (13), conditioning on $\overrightarrow{N}_{j+1:k}$, we get

$$\begin{aligned} & \sum_{\overrightarrow{n}_{j+1:k}} \{\Pr(N_j \geq \bar{n}_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}, \overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}) \\ & \qquad \Pr(\overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k} | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k})\} \geq \\ & \sum_{\overrightarrow{n}_{j+1:k} \in S_{\bar{n}_j}} \{\Pr(N_j \geq \bar{n}_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}, \overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}) \\ & \qquad \Pr(\overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k} | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k})\}. \end{aligned} \quad (16)$$

Note that for $\overrightarrow{n}_{j+1:k} \in S_{\bar{n}_j}$,

$$\Pr(N_j \geq \bar{n}_j | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}, \overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}) = 1.$$

So combining (15) and (16), if we can show that

$$\Pr(\overrightarrow{N}_{j+1:k} \in S_{\bar{n}_j} | \overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}) \geq \Pr(\overrightarrow{N}_{j+1:k} \in S_{\bar{n}_j}),$$

then we would have proven (13). Using Bayes' formula we can rewrite the above inequality as,

$$\Pr(\overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k} | \overrightarrow{N}_{j+1:k} \in S_{\bar{n}_j}) \geq \Pr(\overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k}).$$

Conditioning again on $\overrightarrow{N}_{j+1:k}$, we get

$$\begin{gathered}\sum_{\overrightarrow{n}_{j+1:k} \in S_{\bar{n}_j}} \{\Pr(\overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k} | \overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}) \\ \Pr(\overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k} | \overrightarrow{N}_{j+1:k} \in S_{\bar{n}_j})\} \geq \\ \sum_{\overrightarrow{n}_{j+1:k}} \{\Pr(\overrightarrow{Z}_{j+1:k} \geq \overrightarrow{z}_{j+1:k} | \overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k}) \\ \Pr(\overrightarrow{N}_{j+1:k} = \overrightarrow{n}_{j+1:k})\}\end{gathered} \tag{17}$$

This is true as a consequence of Lemma 2.3. ■

REFERENCES

[1] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijaykumar, and P. Whiting. CDMA data QoS scheduling on the forward link with variable channel conditions. *Bell Laboratories Technical Report*, Apr. 2000.

[2] M. Andrews, L. Qian, and A. L. Stolyar. Optimal utility based multi-user throughput allocation subject to throughput constraints. In *INFOCOM 2005. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2005.

[3] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi. CDMA-HDR: A bandwidth-efficient high-speed wireless data service for nomadic users. *IEEE Communication Magazine,*, pages 70–77, July 2000.

[4] T. Bonald. A score-based opportunistic scheduler for fading radio channels. In *Proc. of European Wireless*.

[5] R. R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn. Statistical service assurances for traffic scheduling algorithms. *IEEE Journal on Selected Areas in Communications*, 18:2651 – 2664, Dec. 2000.

[6] G. C. Fox, R. D. Williams, and P. C. Messina. *Parallel Computing Works*. Morgan Kaufmann Publishers, 1994.

[7] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo*, volume 3, pages 1854 – 1858, May 2000.

[8] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia. Exploiting medium access diversity in rate adaptive wireless lans. In *Proc. of the 10th annual international conference on Mobile computing and networking*, pages 345 – 359, Sept. 2004.

[9] E. Knightly and N. B. Shroff. Admission control for statistical QoS: Theory and practice. *IEEE Network*, 13:20 – 29, Mar. 1999.

[10] R. Knopp and P. Humblet. Information capacity and power control in single cell multi-user communications. In *Proc. IEEE International Computer Conference*, volume 1, pages 331 – 335, June 1995.

[11] D. Park, H. Seo, H. Kwon, and B. G. Lee. A new wireless packet scheduling algorithm based on the cdf of user transmission rates. In *Proc. IEEE Globecom*, pages 528–532, November 2003.

[12] D. Park, H. Seo, H. Kwon, and B. G. Lee. Wireless packet scheduling based on the cumulative distribution function of user transmission rates. *to appear in IEEE Transactions on Communications*, 2005.

[13] S. Patil. Opportunistic scheduling and resource allocation among heterogeneous users in wireless networks, Ph.D. thesis, Univeristy of Texas at Austin. *available at http://www.ece.utexas.edu/˜ patil/Thesis.pdf*, 2006.

[14] S. Patil and G. de Veciana. Measurement-based opportunistic scheduling for heterogeneous wireless systems. In *Submitted for journal publication, available at http://www.ece.utexas.edu/˜ patil/measurement.pdf*.

[15] X. Qin and R. Berry. Opportunistic splitting algorithms for wireless networks with heterogeneous users. In *Proc. Conference on Information Sciences and Systems (CISS)*, March 2004.

[16] T. S. Rappaport. *Wireless Communications, Principles and Practice*. Pearson Education, 2002.

[17] S. Shakkottai and A. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in HDR. In *Proc. of the 17th International Teletraffic Congress (ITC-17), Salvador da Bahia, Brazil*, September 2001.

[18] S. Shakkottai and A. Stolyar. Scheduling for multiple flows sharing a time-varying channel: The Exponential rule. *American Mathematical Society Translations, Series 2, A volume in memory of F. Karpelevich, Yu. M. Suhov, Editor*, 207, 2002.

[19] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48:1277 – 1294, June 2002.

[20] D. Wu. Providing quality-of-service guarantees in wireless networks, Ph.D. thesis, Carnegie Mellon University. Aug. 2003.

[21] D. Wu and R. Negi. Effective capacity: A wireless link model for support of quality of service. *IEEE Transactions on Wireless Communications*, 2:630–643, July 2003.

[22] D. Wu and R. Negi. Downlink scheduling in a cellular network for quality-of-service assurance. *IEEE Transactions on Vehicular Technology*, 53:1547–1557, Sept. 2004.

Shailesh Patil received his Bachelor in Electronics & Communications from Delhi University, India, in 2001 and M.S. and Ph.D. both in Electrical & Computer Engineering from University of Texas at Austin in 2004 and 2006 respectively. His research interests include cross-layer design, scheduling of users and provding quality of service in wireless networks. He is a recipient of Texas Telecommunications Engineering Consortium (TxTEC) Fellowship in 2002.

Gustavo de Veciana (S88-M94-SM 2001) received his B.S., M.S., and Ph.D. in Electrical Engineering from the University of California at Berkeley in 1987, 1990, and 1993 respectively. He is currently a Professor at the Department of Electrical and Computer Engineering at the University of Texas at Austin. His research focuses on the design, analysis and control of telecommunication networks. Current interests include: measurement, modeling and performance evaluation; wireless and sensor networks; architectures and algorithms to design reliable computing and network systems. Dr. de Veciana has been an editor for the IEEE/ACM Transactions on Networking. He is the recipient of General Motors Foundation Centennial Fellowship in Electrical Engineering and a 1996 National Science Foundation CAREER Award, co-recipient of the IEEE William McCalla Best ICCAD Paper Award for 2000, and co-recipient of the Best Paper in ACM Transactions on Design Automation of Electronic Systems, Jan 2002-2004.