Minimizing file transfer delays using SRPT in HSDPA with terminal constraints

Jari Melasniemi, Pasi Lassila (corresponding author), Samuli Aalto
Aalto University School of Science and Technology, Otakaari 5, FIN 02150, Finland
Email: firstname.lastname@tkk.fi

***Abstract*—In an HSDPA system, multiple users are scheduled in a time slot due to constraints on the user terminals with respect to how many codes a particular user can utilize. We model the allocation of the codes at the so-called flow level. This results in a particular multiserver queuing model, where codes correspond to servers and multiple servers are allocated per flow subject to constraints on the maximum number of codes. In this context, we focus on minimizing the mean flow delays by utilizing flow-level information on the remaining service times. While SRPT is the optimal policy for minimizing the mean delay in an M/G/1 queue, no such optimality results exist for the dynamic setting in multiserver models. We derive a heuristic SRPT policy for the system and evaluate its performance against the fair round-robin policy, which can be modeled at the flow-level as a processor sharing system. The results demonstrate that using SRPT-like scheduling can significantly decrease the overall mean delays, as well as the conditional delays.**

***Index Terms*—flow-level analysis, multiserver models, size-based scheduling, SRPT, cellular networks, HSDPA**

I. Introduction

Modern cellular systems (HSPA, LTE) allow the radio resources to be scheduled very flexibly. The scheduling interval is also very short (in the order of milliseconds) and the scheduler has state information allowing the selection of the coding and modulation to match the state of the users' channels. These properties are especially useful for optimizing the performance of elastic data traffic, which roughly corresponds to file transfers controlled by TCP.

Performance of elastic traffic in a given system manifests itself at the so-called flow-level, where the system must be considered in a dynamic setting with random-sized flows arriving stochastically. Much of the modeling and analysis concentrates on the case where a single user is scheduled at a time (e.g., as in 1xEV-DO systems). In a single cell setting and assuming that the scheduler is not able to exploit the instantaneous rate variations across the users (non channel-aware scheduling), the fair round-robin scheduler can be modeled as the classical single server M/G/1 processor sharing (PS) queue, see [1]. Multicell configurations with full frequency reuse [2] or various frequency reuse patterns [3] have also been studied with PS models, but the base stations have been assumed to operate independently, which simplifies the modeling. A multiuser scheduling problem has been studied in the context of HSUPA in [4], where different ways to implement the round-robin scheduling has been modeled at the flow-level using processor sharing models. However, the above works do not consider the additional impact of using information on the flow sizes to further minimize the delays.

In this paper, we consider the impact of this additional information on a particular multiuser scheduling problem. More specifically, we consider the performance of elastic traffic in the single cell scenario, where the users are randomly located around the base station. Given the location of the user, the achievable service rate is assumed to be constant, i.e., all random variations are assumed to average out at the flow time scale. Instead of assuming single user scheduling, we consider a system where multiple users are scheduled simultaneously. This happens for example in HSDPA, where due to the terminal constraints limiting the maximum number of codes a user can utilize, the system needs to schedule simultaneously many users in order to make use of all available codes.[1] This gives rise to a multiserver scheduling problem, where the codes can be viewed as servers and the terminal constraints place limitations on how the codes can be allocated. In practice, the system supports different terminal categories (i.e., users have a different limitation with respect to the maximum number of codes), which are modeled as classes.

As discussed earlier, in the single cell/single user scheduling setting the system corresponds to an M/G/1 queue. In this case it is well known that SRPT (Shortest Remaining Processing Time) policy is optimal for minimizing the mean flow delay [5]. The performance of SRPT policy and also non-anticipating policies in the M/G/1 single cell wireless model have been studied in [6]. For multiserver problems, such SRPT optimality results do not exist. What is known is that the so-called SRPT-FM (SRPT Fastest Machine) is optimal in the static setting, where new flows do not arrive and one server is allocated per each flow (so-called standard multiserver model), see [7].

We first consider the homogeneous case, where all users have the same terminal category. Then our model corresponds to the standard multiserver model. We compare the performance of the fair round-robin policy against the SRPT policy. The performance of the round-robin policy can be evaluated analytically from the standard multiserver PS queue, while the SRPT policy has been simulated. For the heterogeneous case, a PS model with weighted sharing between the classes can be derived for exponentially distributed service times, but for general service times, the system needs to be simulated, in order to evaluate the mean flow delay. However, we show nu-

[1] In an OFDMA system, the codes correspond essentially to carriers.

merically that the sensitivity is not that great. We additionally derive an SRPT policy that applies the principle that all servers (codes) must be allocated and at any given time those flows are served that have the smallest remaining service time, while taking into account the restrictions on the maximum number of codes per user (terminal constraints). This requires a dynamic ranking of the users. By simulations we show that SRPT gives a significant gain in the mean delay over PS scheduling (both in the homogeneous and heterogeneous settings). We additionally study the unfairness (cf., unfairness of SRPT in M/G/1 [8]) with respect to the classes, as well as conditioned on size and rate. The results show that SRPT does not increase unfairness between classes too much compared with PS, and the conditional delays of even the longest service times (large size and low rate) can be lower under SRPT than PS.

Note that we are in this paper focusing on the case where the scheduler does not utilize instantaneous channel state information. This assumption is valid when the channel state can not be estimated sufficiently accurately, e.g., when users are moving fast. In channel-aware scheduling, the scheduler is aware of the instantaneous rate variations across different users and can allocate the channel to the user with the best channel condition at that moment. Under certain assumptions these schedulers can also be modeled using processor sharing models with state-dependent service rates [9]. The available results on the flow-level properties of channel-aware schedulers are related to their stability properties, see [9], [10], [11]. Recently there has been some progress in analyzing delay optimal channel-aware schedulers using SRPT-like information on the remaining sizes, see [12]. It has been shown that when the so-called capacity region is bounded by a polymatroid, under this polymatroid capacity region the optimal static policy can be mapped to the SRPT-FM policy.

The paper is organized as follows. The model is introduced in Section II. The homogeneous and heterogeneous cases are treated in Sections III and IV, respectively. Conclusions are given in Section V.

II. System model and scheduling policies

In this section we describe our modeling assumptions. The objective is to establish a flow-level model of the HSDPA system that incorporates the impacts of user terminal constraints on the scheduling.

A. Traffic/user model

Consider the transmission of downlink traffic from the base station to the users in a single cell. The traffic consists of elastic flows representing, e.g., file transfers. Each flow is associated with a particular user, and thus subsequently we may use the words user/flow interchangeably. We consider a spatial model where the users are assumed to be uniformly distributed in the cell. The cell itself is modeled as a unit circle. Thus, the distance R from the base station to the receiving user has the cumulative distribution function

$$\mathrm{P}\{R \le r\} = r^2, \quad r \le 1.$$

The flows are grouped into classes depending on the user's terminal constraints which limit their processing capabilities (i.e., all flows in the same class have the same constraints). The allocation of resources to the flows depends on the class, as will be discussed later. Class-i flows, $i = 1, \dots, I$, arrive at the base station according to a Poisson process with rate λ_i and the flow sizes X [bit] are assumed to be independent and identically distributed random variables obeying the density $f_X(x)$ (same density for all classes).

B. Channel model

For the achievable rate of a user at distance r from the base station, we use the following model. We consider the system in the limit where the time-slot duration is negligible compared with the flow sizes (i.e., we have time scale separation between the flow level and the time-slot level). Let $c(r)$ [bit/s] denote the total transmission rate at which the base station can serve a user at distance r, if all radio resources are allocated to the user and the user is alone in the system. We may for example use the model (see [1], [6])

$$c(r) = \begin{cases} c_0, & r \le r_0, \\ c_0 \left(\frac{r_0}{r}\right)^{\alpha}, & r > r_0, \\ 0, & r \ge 1, \end{cases} \tag{1}$$

where c_0 is the maximum data rate of the system and α, called the attenuation factor, typically takes values in the range from 2 to 4. The received power attenuates due to path loss and the rate is linear in the received power. Note that the above is just an example and $c(r)$ can easily be modified to take into account more complex interference phenomena. The important property of $c(r)$ is that at a given location r, the transmission rate $c(r)$ is constant, i.e., $c(r)$ models the mean transmission rate that is available to a particular user using all radio resources.

In the HSDPA system, due to its underlying CDMA-based technology, the radio resources that are allocated to the users are CDMA codes. In the system, there are altogether K codes that can be shared between the users transmitting elastic traffic. We assume that K does not change over time, or changes at such a slow time scale that the flow-level time scale can be considered roughly stationary over the time scale at which K changes.

Associated with each user/flow is the maximum number of codes that the user's terminal can handle, the so-called terminal category. In the system, there is a finite set of possible values for the maximum number of codes a terminal can use, e.g., in HSDPA terminals can use 5, 10 or 15 codes. In our model, those users that have the same terminal category belong to the same class, and we denote by k_i the number of codes a class-i user can utilize.[2] The K codes are assumed to be orthogonal. Thus, for a given user at distance r from the base

[2]In the HSDPA system, terminal category includes also parameters related to the supported coding schemes, in addition to the maximum number of codes. Including these parameters in our model is straight forward but adds to the number of classes.

station, the rate per code is $c(r)/K$. Conversely, when a user belonging to class i is allocated k_i codes, the rate is $k_i c(r)/K$.

In queuing terminology we can interpret each code as a server. The task of the scheduler is then to allocate the servers to flows. When the scheduler can always allocate all codes/servers to a single flow, the system corresponds to the M/G/1 queue, see [1], [6]. However, when we take into account the limitation that a class-i flow can not use more than k_i servers, the system may need to serve multiple users in parallel to keep all servers busy, i.e., the system corresponds to a complex multiserver system.

The load of class-i flows, ρ_i, is given by

$$\rho_i = \lambda_i \mathrm{E}[S_i],$$

where S_i denotes the random variable for the service time requirement of class-i flows. For a class-i flow of size X at a random distance R, we have that $S_i = X/(k_i c(R)/K)$. Since X and R are independent, it follows

$$\mathrm{E}[S_i] = \mathrm{E}\left[\frac{X}{k_i c(R)/K}\right] = \frac{K}{k_i}\mathrm{E}[X]\,\mathrm{E}\left[\frac{1}{c(R)}\right].$$

Assuming that the scheduler uses any work conserving policy, the system is stable if

$$\sum_{i=1}^{I} \rho_i < 1.$$

Within the stability limit we can optimize the performance of the system by using different scheduling policies.

C. Scheduling policies

To study the performance of the system, our main focus is on minimizing the mean flow delay (i.e., the mean sojourn time) by utilizing information about the flow sizes. To this end, the following two policies are considered.

PS: Round robin represents a fair baseline policy, where size information is not used at all. At the flow-level, round-robin policy can be modeled as a processor sharing queue. In certain cases, the mean flow delay in the PS model can also be analyzed explicitly. Thus, in this paper, when referring to the round-robin policy we speak about the corresponding PS model.

SRPT: In size-based scheduling the underlying idea is to favor smaller flows in order to efficiently minimize the number of flows in the system, which also minimizes the mean flow delay. In the M/G/1 queue (i.e., under single user scheduling), SRPT is a preemptive policy where the server at any moment in time always serves the flow with the least amount of remaining service time. It yields the optimal policy for minimizing the mean flow delay, see [5]. For the standard multiserver model (one server for each flow, server speeds may be heterogeneous) the so-called SRPT-FM is optimal for minimizing the mean flow delay in the static setting with no new arrivals, see [7, Theorem 5.4.2]. However, no such optimality results exist for the dynamic setting with stochastically arriving flows. In this paper we develop an SRPT-like policy for the multiserver model of HSDPA resource allocation with terminal constraints, which requires a modification of the SRPT-FM policy to account for the fact in our model several servers can be allocated for each job.

III. Homogeneous users

Next we consider a setting where all users are homogeneous. In this case, the model can be mapped to the standard multiserver model.

A. Analysis of the schedulers

Assume that all users are identical with respect to the terminal constraints, i.e., the maximum number of codes that a user can utilize is the same, $k_i = k$, for all users. Assuming further that K/k is integer valued, we can take $m = K/k$ to mean the number of servers in the standard multiserver model (with one server per each flow). In the dynamic setting with stochastically arriving flows, the system corresponds to the M/G/m queuing model.

PS: The fair round-robin policy can be modeled as a PS system. Assume first that the file sizes X obey the exponential distribution, that is we consider the M/M/m-PS queue, where the service rate of each server is denoted by $\mu = 1/\mathrm{E}[S]$ and the load is defined as

$$\rho = \lambda/(m\mu).$$

In this case, the stochastic process for the number of users in the system $N(t)$ evolves as in a birth-death process with, for state n, an upward transition rate λ and downward transition rate $\min(n, m)\mu$. The equilibrium distribution of the process can be obtained easily from which we directly get the mean number customers $\mathrm{E}[N]$. Standard results also yield the conditional mean delay of a flow of size x at distance r from the base station, $\mathrm{E}[T(x, r)]$,

$$\mathrm{E}[T(x, r)] = \frac{x}{c(r)}\,\frac{\mathrm{E}[N]}{\rho}.$$

The M/M/m-PS system is also insensitive to the distribution of the service times [13]. Thus, the above results also hold for the more general M/G/m-PS model.

SRPT: For the standard multiserver model, SRPT-FM is a preemptive policy where the fastest machine always serves the flow with the smallest remaining service time, i.e., the scheduler is always serving in parallel those n flows that have the smallest remaining service times. Note that the scheduler assumes exact knowledge of the remaining service time. In our model, this implies exact knowledge of both the remaining amount of bits to transmit, as well as the rate $k_i c(r)/K$ at which the user is served. In our case, all servers serve the flows at the same rate, i.e., the servers have uniform rates, and the size and the random location only affect the service time of the flow. Since the server rates are uniform, we here refer to SRPT-FM simply as SRPT. Due to lack of optimality or analytical results on SRPT in the M/G/m queue, we need to use simulations to estimate the performance.

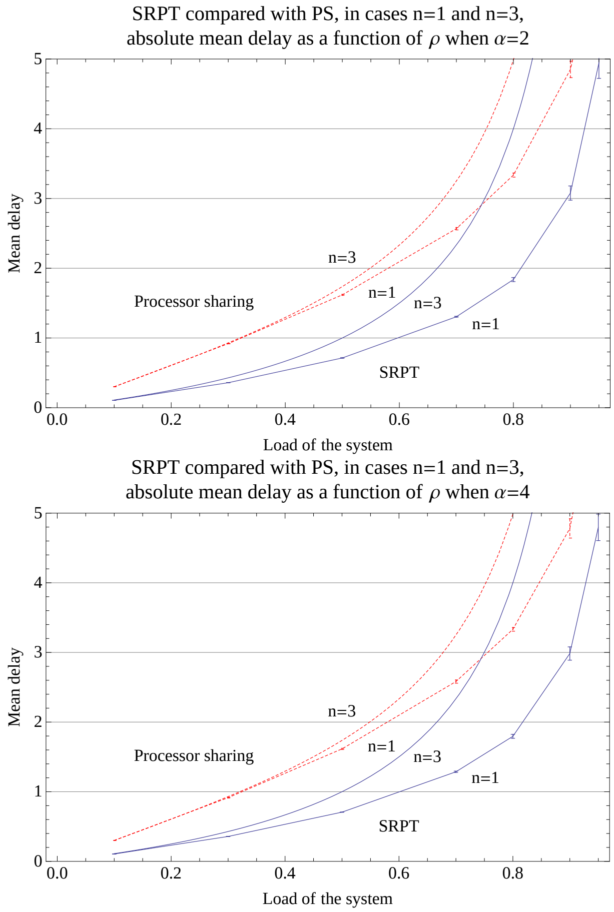

Fig. 1. The absolute values of the mean delay for SRPT and PS with $\alpha = 2$ (upper panel) and $\alpha = 4$ (lower panel).

B. Numerical results

Next we evaluate the performance gains that can be achieved by applying SRPT compared with PS. The following parameters are fixed in the numerical studies. In the rate function $c(r)$, $c_0 = 1$ and $r_0 = 1/7.94$ (see [1], [6]). The flow sizes are exponentially distributed and the flows arrive with rate $\lambda = 1$ [1/s]. To change the load ρ, we modify the mean flow size $\mathrm{E}[X]$. For a given data point, the simulation consists of 10 repeated runs, each with approximately $2 \cdot 10^5$ arrivals, of which 10^5 have been excluded from the statistics collection due to the initial transient.

We first study the absolute mean flow delays under PS and SRPT for $n = 1$ and $n = 3$ servers for $\alpha = 2$ and $\alpha = 4$. The results are shown in Figure 1 with $\alpha = 2$ (upper panel) and $\alpha = 4$ (lower panel). Different values of the load of the system (ρ) are drawn as their own lines. Cases with $n = 3$ are shown with dashed lines. It can be seen from the figure that the performance improvement of the SRPT policy is most notable with high values of ρ. With low values of ρ there is not much difference between PS and SRPT. However even in that case there is a distinguishable difference between the policies in favor of SRPT. Path loss exponent affects the transmission rate function and thus changes the service time distribution, but the effect seems to be minimal for SRPT. Note that the performance of PS is insensitive to the service time distribution.

Based on Figure 1 the performance improvement of the SRPT policy seems to be decreasing when the number of servers, or codes, increases. This aspect will be examined further next. The case has been simulated with different values of ρ and $m = K/k$ is increased from 1 to 25. Figure 2 shows the mean delay ratio obtained with SRPT to the corresponding value under PS. From the figure it can be seen that the performance gain which can be achieved with the SRPT policy is largest when only a few servers are used and ρ is relatively high. With smaller values of ρ, the advantage of SRPT decreased to almost zero already with values of n less than ten. With high loads, there is still a remarkable improvement even if the value of n is more than 20.

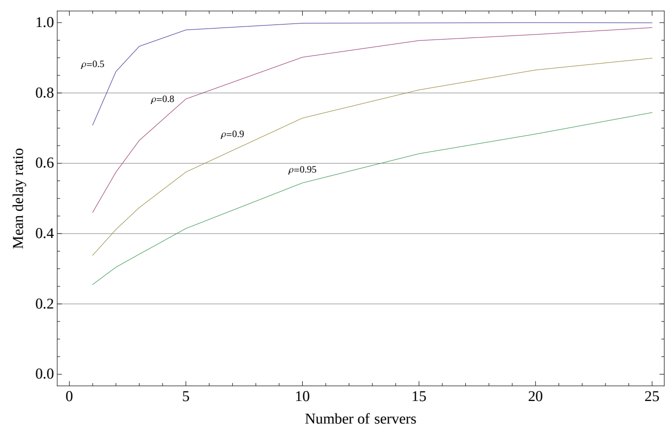

Fig. 2. The effect of $n = K/k$ on the performance improvement of SRPT.

Recall that SRPT requires exact knowledge of both the remaining amount of bits and the rate of the user. Now the policy is changed so that it will prioritize the flows for which the bit sizes are the lowest without using any rate information. Figure 3 shows the delay ratio of SRPT to PS and the delay ratio of the modified SRPT to PS for $n = \{1, 2, 3\}$ with $\alpha = 2$. The performance of the modified SRPT policy seems to be much worse than the basic SRPT. With load values more than 0.5 the modified version is even worse than the PS policy. The variation in different policies is smaller with low values (below 0.5) of ρ. With higher values of ρ the basic SRPT outperforms the modified version systematically. It is not worth selecting the flows by the bit size. If the transmission rate is not known and it is hard to measure or determine, it is better to use PS than this kind of a modified version of SRPT.

IV. Heterogeneous users

In the general model, flows are grouped in classes according to their terminal constraints, that is class-i flows can use k_i codes. It turns out that the round-robin policy can be analyzed as a PS system under Markovian assumptions. Additionally, we introduce the generalized version of SRPT-FM for our setting.

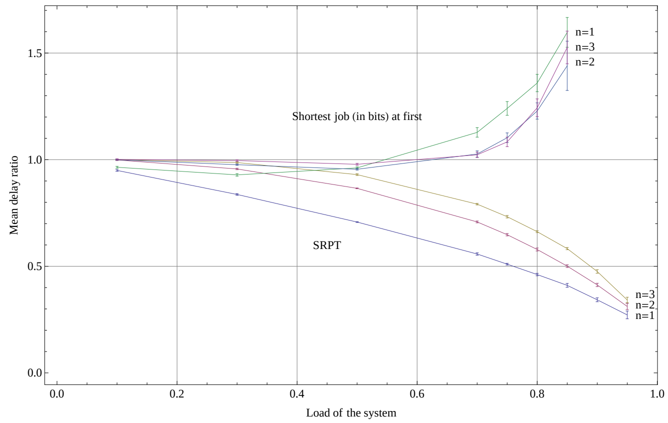

Fig. 3. Mean delay of SRPT and modified SRPT compared with PS as a function of the load of the system with 95 % confidence interwalls.

A. PS model and its analysis

In the round-robin policy, the scheduler is at the time slot level allocating the codes to the users in a cyclic order. We denote by n_i the number of flows in class-i. Also, assume that the service time requirement is exponentially distributed and let $1/\nu$ denote the mean service time when all K codes are allocated to a single flow,

$$\frac{1}{\nu} = \mathrm{E}[X]\,\mathrm{E}\left[\frac{1}{c(R)}\right].$$

If $\sum_i k_i n_i \le K$ then clearly each flow gets all the requested k_i codes and the fraction of the service rate of class-i flows, given that there are n_i class-i flows, equals $(k_i n_i/K)$. When $\sum_i k_i n_i > K$ the time to go through all flows is greater than the time slot length. Our assumption is that round robin can be implemented in an ideal manner so that the round can continue over multiple time slots and then the service pattern repeats itself. Additionally, our time scale separation assumption implies that this cyclic round-robin is performed infinitely many times until the next departure/arrival occurs. Thus, we can take $\sum_i k_i n_i$ to represent a "pseudo-frame" within which a fraction $k_i n_i / \sum_j k_j n_j$ is allocated to the class-i flows. Within the class, the flows share the time according to pure PS. Thus, we can represent the system by a multidimensional Markov process where in state $\mathbf{n} = (n_1, \ldots, n_I)$ the transition rates are given by

$$\begin{cases} \mathbf{n} \to \mathbf{n} + \mathbf{e}_i : \lambda_i, \\ \mathbf{n} \to \mathbf{n} - \mathbf{e}_i : \frac{k_i n_i}{\max(K, \sum_j k_j n_j)}\nu, \end{cases}$$

where $\mathbf{e}_i$ denotes a vector with 1 in the i^{th} element and all others are 0. Note that letting $K = 1$ and $k_i \ge 1$, for all i, corresponds to the standard Discriminatory Processor Sharing (DPS) queue. Also, if $k_i = 1$, for all i, the model represents the standard (multiclass) PS queue with K servers. Thus, we call the model a multiserver DPS queue. The steady state solution of the system can be solved by standard numerical techniques, from which we can obtain $\mathrm{E}[N]$ (as well as $\mathrm{E}[N_i]$).

The model above is sensitive, i.e., it holds only for the case when service times are exponential. In reality this is not a natural assumption as the service time distribution is jointly defined by the flow size distribution and the distribution of the rates (that depends on the distribution of the users). In the non-Markovian setting, i.e., when flow sizes and rates are determined by their own respective distributions, our flow-level model for the service rate of a user still applies. A particular user at distance r from the base station is served at the rate

$$\frac{k_i}{\max(K, \sum_j k_j n_j)} c(r).$$

This yields a very efficient way to simulate the system with arbitrary flow size distributions and rate functions directly at the flow-level, instead of having to implement the round-robin scheduler in slotted time.

Despite the sensitivity of the Markov model to the service time distribution, it is quite common in processor sharing models that the sensitivity may not be that great when considering the mean flow delays. To illustrate this, we show in Figure 4 a comparison with the results from the Markov model (solid lines) and simulations obtained with exponential file sizes (dashed lines) and Pareto file sizes with shape parameter $\beta = 2$ (dotted lines). The figure shows the mean number of class-1 and class-2 flows as a function of the load in a system where $\lambda_1 = \lambda_2 = 0.5$, $K = 75$, $k_1 = 5$ and $k_2 = 10$. In the rate function $c(r)$, we used $\alpha = 2$. The simulations consisted of $3 \cdot 10^5$ arrivals for the exponential case and $5 \cdot 10^5$ for the Pareto case, of which, in both cases, 10% were ignored in the data collection due to the initial transient. As can be seen in Figure 4, the different lines practically coincide showing the near-insensitivity of the system.

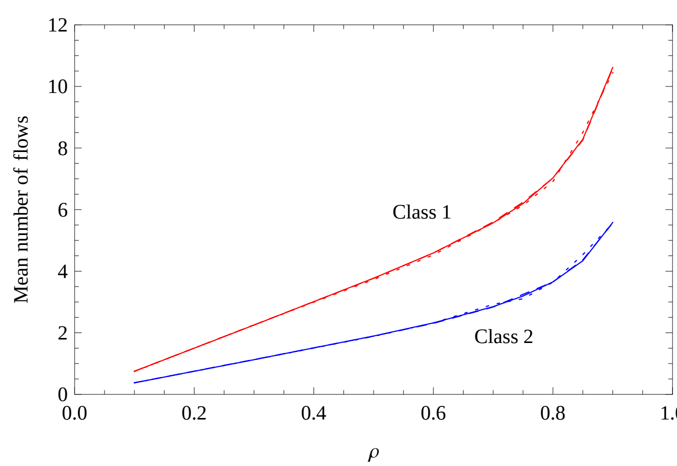

Fig. 4. Mean number of class-1 and class-2 flows as a function of ρ in the Markovian model (solid lines) compared with exponential (dashed lines) and Pareto (dotted lines) distributed sizes.

B. SRPT policy and its analysis

The SRPT-FM policy can not be directly applied in the heterogeneous case. To this end we derive a heuristic SRPT policy, which attempts to maintain the principle that at each scheduling instant, those flows are taken into service that have the shortest remaining processing time. We restrict the scheduling instants to be the arrival and departure instants (similarly as in SRPT-FM for the standard multiserver model). The main difference is that, while in the standard multiserver case exactly one server is allocated to each flow, in the heterogeneous case not all class-i flows taken into service will get exactly all k_i codes since for the last selected flows the remaining codes may be less than the required k_i. However, our heuristic rule will still allocate all the remaining codes to some flow according to the SRPT principle of serving the smallest flow in time. This is achieved by ranking the flows dynamically at each scheduling instant.

Let L denote the number of already allocated codes at an arbitrary scheduling instant (departure/arrival), i.e., $K - L$ codes are remaining. The allocation of codes to the flows is done in rounds until no unallocated codes remain. At each round the set of flows that are not yet scheduled are indexed by u and the remaining service time of u is computed by dividing the remaining bits with

$$\frac{\min(K - L, k_{i(u)})}{K} c(r(u)),$$

where $i(u)$ denotes the class index of flow u and $r(u)$ is the distance r of flow u from the base station. Then the scheduler selects the flow that had the smallest remaining service time and the number of remaining codes L is updated accordingly.

Note that it is easy to see that the above policy is not optimal in the static setting for minimizing the flow delay. Consider the following case with 2 classes having $k_1 = 1$ and $k_2 = 2$, and there are two servers both with a service rate 1. At time 0 there are 2 flows in class 1 with sizes $1+\epsilon$ and 1 flow in class 2 with $k_2 = 2$ and size 2. Then our heuristic SRPT policy will first serve the flow from class 2, which gets a delay of 1 time unit and then the two flows from class 1 simultaneously that finish their service at time $2 + \epsilon$. This gives a total delay of $5 + 2\epsilon$. However, the optimal policy in this case is to first serve the two flows from class 1 that will both finish at time $1 + \epsilon$, after which the flow from class 2 will be served so that it completes at time $2 + \epsilon$. This gives the optimal delay of $4 + 3\epsilon$. Thus, the optimal policy is clearly more complex than our heuristic even in the static setting. Despite this the proposed heuristic may still be close to optimal in the dynamic setting. Next we demonstrate its performance via simulations.

C. Numerical results

Here we consider simulation results for the heterogeneous case with two classes having $k_1 = 5$ and $k_2 = 15$ and $\lambda_1 = \lambda_2 = 0.5$. The total number of codes $K = 20$. With respect to k_i and K this case can be considered realistic. [3] The rate function parameters are $\alpha = 2$, $c_0 = 1$, and $r_0 = 1/7.94$. The file sizes are drawn from an exponential distribution. For each simulated value of the load, the simulations were repeated 10 times with approximately $2 \cdot 10^5$ arrivals in each run of which 10^5 are ignored in the data collection due to the initial transient. In this section, when discussing the results related to the round-robin policy we refer to it as PS, and similarly the heuristic SRPT policy is just referred to as SRPT.

We first study the overall mean delay performance. The results are presented in Figure 5. It can be seen that SRPT improves greatly the performance compared to PS. The gains are of the same order as with the homogeneous case with a small number of servers, cf. Figure 2. This is to be expected, as with the parameters that we are using in our heterogeneous setting the degree of multiuser scheduling is relatively small (class-2 flows can use almost all codes) and the system is not that different from the homogeneous case with a small number of servers (small value for the ratio K/k). The figure also depicts how different classes behave, which shows that SRPT does not dramatically increase the unfairness between the classes compared with PS. Simulations with $K = 75$ codes, where the degree of multiuser scheduling increases, the benefits of SRPT started showing at a higher value of the load, again similarly as in the homogeneous case with higher number of servers. For details and more simulation results, see [14].

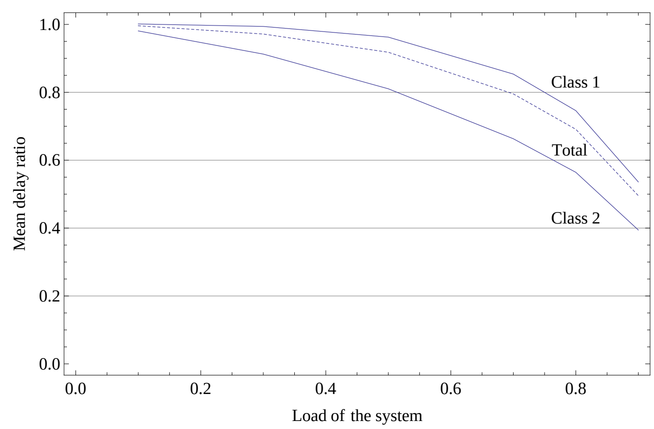

Fig. 5. Mean delay of SRPT compared with PS with $K = 20$.

To get more insight on the impact of SRPT scheduling on the delay properties, we examine the conditional mean delays with respect to the distance from the base station and the flow size. Both variables are divided into two categories which leads to a total of four cases: small flows near the base station, small flows far from the base station, large flows near the base station, and large flows far from the base station. As a threshold between small and large flows, the mean flow size $E[X]$ has been used. As a threshold between the distances far and near, the radius r_0 has been used, as summarized in Table I.

The performance improvement of the SRPT policy for the classified data is depicted in Figure 6. The mean delay of the SRPT in each category (small flows near, small flows far, large flows near, and large flows far) have been compared with the

[3]The values $k_i = \{5, 10\}$ correspond to actual values used in HSDPA terminals and also typically the total number of codes $K = 15$.

TABLE I
CLASSIFICATION PARAMETER SET

Name of the class	Value range
Small	$x \in [0, \mathrm{E}[X]]$
Large	$x > \mathrm{E}[X]$
Near	$r \in [0, r_0]$
Far	$r \in [r_0, 1]$

PS policy. It can be seen from Figure 6 that the performance improvement is the lowest in the large flows far from the base station. The flows near the base station, regardless of the size of the flow, seem to benefit the most from the SRPT. Small flows far from the base station can benefit almost as much. The performance improvements in those flows are much more significant than the improvements in the bigger flows in the same distance range. These results are well in line with the fact that SRPT favors small jobs and larger ones will suffer relatively. However, Figure 6 shows that with high loads even large flows far from the base station can benefit from SRPT compared with PS. We also tested with a classification set, where the threshold for large files was $2\mathrm{E}[X]$ and for the distance $2r_0$ (i.e., we are looking even more into the tail of the service times) and the conclusion was essentially the same, see [14] for details.

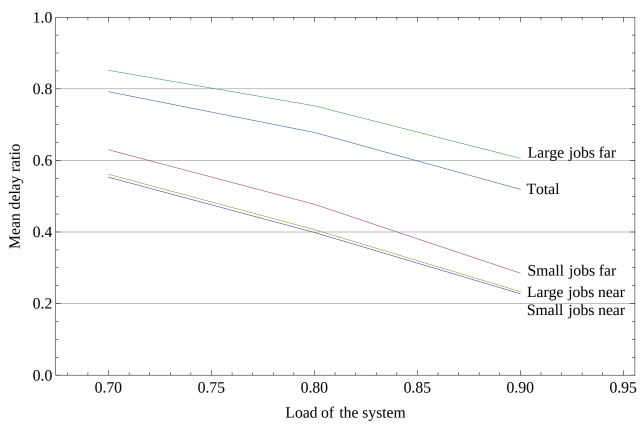

Fig. 6. Relative performance improvement of SRPT compared to PS for the conditional mean delays.

Finally we look at the mean conditional mean delays in the different categories also between the classes. The results are shown for $\rho = 0.7$ for SRPT in Table II and for PS in Table III. It can be seen from these results that the mean delay ratio between the classes 1 and 2 (the third column) does not depend much on the classification of the data. The ratio of the delay of the classes seems to be very near the ratio of the number of codes for the classes, and the ratio is only slightly higher than for PS. Also, we can observe that the mean delays are indeed smaller for SRPT in every category for both classes, i.e., everyone seems to benefit. Changing the classification thresholds to $2\mathrm{E}[X]$ and $2r_0$ did not change this conclusion, see [14].

TABLE II
CONDITIONAL MEAN DELAY RESULTS FOR SRPT ($\rho = 0.7$, $K = 20$).

	Mean delay	Mean delay of Class 1/Class 2
Class 1	0.0654 ± 0.0017	
Class 2	0.0219 ± 0.0003	
Small near, total	0.0436 ± 0.0009	2.9879
Class 1	2.3465 ± 0.0082	
Class 2	0.7872 ± 0.0019	
Small far, total	1.5668 ± 0.0047	2.9807
Class 1	0.1003 ± 0.0024	
Class 2	0.0344 ± 0.0005	
Large near, total	0.0673 ± 0.0010	2.9154
Class 1	4.7958 ± 0.0304	
Class 2	1.6587 ± 0.0069	
Large far, total	3.2273 ± 0.0182	2.8912
Total	2.5911 ± 0.0120	

TABLE III
CONDITIONAL MEAN DELAY RESULTS FOR PS ($\rho = 0.7$, $K = 20$).

	Mean delay	Mean delay of Class 1/Class 2
Class 1	0.1121 ± 0.0036	
Class 2	0.0458 ± 0.0005	
Small near, total	0.0790 ± 0.0017	2.4466
Class 1	3.5046 ± 0.0133	
Class 2	1.4717 ± 0.0088	
Small far, total	2.4881 ± 0.0081	2.3812
Class 1	0.1718 ± 0.0056	
Class 2	0.0682 ± 0.0026	
Large near, total	0.1200 ± 0.0032	2.5169
Class 1	5.2384 ± 0.0322	
Class 2	2.3405 ± 0.0191	
Large far, total	3.7895 ± 0.0246	2.2381
Total	3.2719 ± 0.0174	

V. CONCLUSIONS

We have studied at the flow level the scheduling of multiple users simultaneously in an HSDPA system by a multiserver model, where the codes correspond to servers and each flow (or user) has a maximum limitation on the number of codes that can be allocated. Also, in the model, users with different maximum code limitations are represented by classes. As a baseline scheduling policy, the fair round-robin policy was used, which can be modeled as a processor sharing system. To minimize the mean flow delay, an SRPT heuristic was derived which at any time ensures that all servers are allocated and that the flows with smallest remaining service times are served subject to the maximum allocation constraints.

Using simulations we compared the gains in the mean delay performance from using SRPT against the corresponding fair PS model. Regarding the overall mean delay, the results demonstrated that for a given level of the load the gain depends on the degree of multiuser scheduling (i.e., on the number of servers). However, for a sufficiently high value of the load, the SRPT heuristic eventually yields significant gains. In a practical system, the benefit can be significant already for moderate loads as the degree of multiuser scheduling is not necessarily that great (cf., the results with $K = 20$ codes). In the study of the conditional mean delays, we saw that, with the categorization that was used with respect to the sizes and the rates, even the longest flows benefited from SRPT compared with PS. Also, the fairness between the classes was not greatly affected by the use os SRPT compared with PS.

The results demonstrated clearly that SRPT can reduce the delays considerably. However, the question of optimality remains still open, even in the static setting.

ACKNOWLEDGEMENT

This research has been partially supported by the AWA (Advances in Wireless Access) project, funded by Ericsson, Nokia-Siemens Networks and TEKES.

REFERENCES

[1] T. Bonald and A. Proutière, "Wireless downlink data channels: user performance and cell dimensioning," in *Proc. of ACM MobiCom*, Sep. 2003, pp. 339–352.

[2] H. van den Berg, R. Litjens, and J. Laverman, "HSDPA flow level performance: the impact of key system and traffic aspects," in *Proc. of ACM MSWiM*, Oct. 2004, pp. 283–292.

[3] T. Bonald and H. N., "Capacity gains of some frequency reuse schemes in OFDMA networks," in *Proceedings of IEEE GLOBECOM*, 2009.

[4] D. C. Dimitrova, H. Van Den Berg, G. Heijenk, and R. Litjens, "Flow level performance comparison of packet scheduling schemes for umts eul," in *WWIC'08: Proceedings of the 6th international conference on Wired/wireless internet communications*, 2008, pp. 27–40.

[5] L. E. Schrage, "A proof of the optimality of the shortest remaining processing time discipline," *Operations Research*, vol. 16, pp. 687–690, 1968.

[6] S. Aalto and P. Lassila, "Impact of size-based scheduling on flow-level performance in wireless downlink data channels," in *Proc. of ITC-20*, Jun. 2007, pp. 1096–1107.

[7] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, 3rd ed. Springer, 2008.

[8] A. Wierman and M. Harchol-Balter, "Classifying scheduling policies with respect to unfairness in an m/gi/1," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 238–249, 2003.

[9] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 636–647, 2005.

[10] S. Borst and M. Jonckheere, "Flow-level stability of channel-aware scheduling algorithms," in *Proc. of WiOpt*, April 2006, pp. 1–6.

[11] S. Aalto and P. Lassila, "Flow-level stability and performance of channel-aware priority-based schedulers," in *Proceedings of 6th Euro-NF conference on Next Generation Internet (NGI 2010)*, 2010, pp. 1–8.

[12] B. Sadiq and G. de Veciana, "Balancing srpt prioritization vs opportunistic gain in wireless systems with flow dynamics," in *Proceedings of 22nd International Teletraffic Congress (ITC-22)*, 2010, pp. 1–8.

[13] J. Cohen, "The multitype phase service network with generalized processor sharing," *Acta Informatica*, vol. 12, pp. 245–284, 1979.

[14] J. Melasniemi, "Size-based scheduling under terminal constraints in cellular systems," Master's Thesis, Aalto University School of Science and Technology, 2010, available at: http://www.netlab.hut.fi/tutkimus/awa/publ/jari_melasniemi_2010.pdf.