Combining Opportunistic and Size-Based Scheduling in Wireless Systems

Pasi Lassila
TKK Helsinki University of Technology
P.O.Box 3000, FI-02015 TKK, Finland
Pasi.Lassila@tkk.fi

Samuli Aalto
TKK Helsinki University of Technology
P.O.Box 3000, FI-02015 TKK, Finland
Samuli.Aalto@tkk.fi

ABSTRACT

HSDPA/HDR systems allow the use of sophisticated opportunistic schedulers that can utilize information on instantaneous channel conditions. On the other hand, for elastic data traffic the size of the files can be used in size-dependent scheduling methods, e.g., the well known SRPT scheduler, to minimize the flow delays. In this paper, we consider the optimal use of both size and channel information for minimizing the flow delay. We derive several heuristics which utilize both types of information. In a static setting with two flows and two rates, the optimal policy can be constructed via dynamic programming and can be compared against the policies using exact size knowledge. In the dynamic setting (stochastically arriving flows with random sizes), extensive simulations have been performed to evaluate the performance of the schedulers under heavy traffic. In the symmetric setting, the differences between the schedulers are clearly visible, while in the asymmetric setting the dynamics are more complex. The results still show that significant gains can be achieved with additionally using size information.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; G.3 [**Probability and Statistics**]: Queueing theory

General Terms

Performance, Algorithms

Keywords

Opportunistic scheduling, size-based scheduling, flow-level modeling, HSDPA/HDR systems

1. INTRODUCTION

Modern 3G cellular networks allow the use of highly sophisticated scheduling mechanisms, and systems such as HSDPA/HDR have been designed specifically for supporting the demands of data traffic. In these systems time is slotted and scheduling decisions can be made at a very small time scale (milliseconds). The idea in the scheduling, in principle, is to allocate the transmission resources of the time slot to a single user at a time, thus effectively eliminating intra-cell interference between the users. The delay tolerance of data applications is higher than for traditional voice traffic, which allows the scheduler more freedom in how to allocate the time slots. This together with the fast time scale of scheduling has given rise to so called opportunistic schedulers, where the scheduler allocates the time slot to the user with currently favorable channel conditions.

We consider the optimal (downlink) scheduling problem from the so called flow-level point of view for a single base station in an HSDPA/HDR cellular network. At the flow level traffic consists of flows, which roughly correspond to file transfers. The flows are elastic meaning that the applications tolerate variations in the instantaneous transmission rate, cf., file transfers controlled by TCP (for FTP and HTTP applications). The setting is also dynamic, where the flows arrive in a stochastic manner and have random sizes. Note that relative to the time slot duration a typical flow requires many time slots to complete. The flow-level corresponds to the time scale at which user's experience the performance of elastic traffic, which is captured by the total time it takes to transmit the whole flow (file), i.e., the flow delay. The task of the scheduler is then to allocate the time slots among the competing flows in an efficient manner, for example to minimize the mean flow delay.

In wireless systems the channel state (or the signal-to-interference-noise ratio) varies randomly, e.g., due to fast fading. At the flow level this implies that the achievable transmission rates vary randomly from time slot to another. However, if the scheduler does not utilize the instantaneous channel state information, at the time scale of flows these rate variations average out, and in the limit when the time slot length goes to zero the system can be modeled as an M/G/1 queue, see [5, 1]. In this context, the processor sharing (PS) queuing discipline approximates the round robin (RR) scheduler. The flow level delays can then be minimized by using flow-level size information, cf., the shortest remaining processing time (SRPT) policy. The benefits of size-based scheduling in wireless systems compared with the plain RR scheduler has been considered in [1].

In the schedulers discussed above, the base station is only at most aware of the mean rate of the channel. So called opportunistic schedulers, on the other hand, try to exploit the random variations in the channel rates in successive time

MSWiM'08, October 27–31, 2008, Vancouver, BC, Canada.

slots to optimize the performance. For example, the proportionally fair (PF) scheduler has been implemented in standardized systems, such as HSDPA and HDR, and always allocates the time slot to the flow that has the highest instantaneous rate relative to its throughput. Indeed, it has been shown that the PF scheduler increases the capacity of the system, see [2, 6].

The idea here is to utilize simultaneously both channel variations and information about the flow sizes, when selecting the user to be served during the next time slot. Even under very simple assumptions about the channel, the optimal scheduling rule is not known.

In the present paper we introduce two classes of schedulers, which utilize the rate and size information in a different manner. In the first class we have so called priority policies that give an absolute priority to the flow with the highest rate. If there are multiple flows with the same highest rate, then among these flows we apply size information, either SRPT-like exact size information or knowledge of service thus far. In the second class we have index policies, which combine the rate and size information into a distinct index value for each flow and the scheduler then selects the flow with the highest index. In this class, we have the TAOS2 scheduler, which was given in [9], serving as the reference policy, as well as the PF scheduler. In a static setting with two flows we can obtain analytical results on the optimal policy by using a dynamic programming approach, and we compare the performance of some of the proposed schedulers against the optimal policy. In the dynamic setting with stochastically arriving flows, extensive simulations of the schedulers under heavy traffic are used to develop insight to their heavy traffic behavior, that is beyond the load region of schedulers that do not utilize channel rate information.

The paper is organized as follows. In Section 2 we compare the performance of the PF scheduler with standard size-based schedulers. The schedulers combining both rate and size information are defined in Section 3. The analytical results on the optimal policy in a static scenario are in Section 4, while the simulation results for the dynamic setting are in Section 5. Finally, the conclusions are in Section 6.

1.1 Related work

As discussed earlier, when rate variations are not utilized by the scheduler, the system can be modeled at the flow level as an M/G/1 queue. For this system the optimal policy minimizing the mean flow delay is given by SRPT, see [13]. Among the non-anticipating policies, the foreground-background (FB) policy, which serves the flow with least amount of attained service, is optimal when the service times belong to the class of distributions with the decreasing hazard rate (DHR) property, see [12, 17]. DHR distributions, such as the Pareto distribution, have been used to model the sizes of Internet flows, see [7]. These ideas were applied to HSDPA/HDR systems in our earlier paper [1]. Here we continue by considering also the use of rate variations and the size information.

The PF scheduler represents the reference scheduler among opportunistic schedulers. Typically the analysis of the PF scheduler, e.g., demonstrating the achievable scheduling gains, has been done in a static setting with a fixed number of users, see, e.g., [8, 16]. Also, so called relative best (RB) schedulers have been analyzed, see [2]. The PF scheduler can however perform in an unfair manner in certain settings, as discussed, for example in [3], where the score-based scheduler is proposed as a solution. In [11], the performance of both PF and score-based approaches are compared, and also the performance degradation due to the need to measure the channel statistics is analyzed.

In this paper the focus is on the flow-level performance. Related work on the round-robin scheduler in HSDPA/HDR systems is in [5], where various system aspects are analyzed from the flow-level point of view. The flow-level performance of weight-based opportunistic schedulers has been analyzed in [6], where it is shown, among other things, that the system under PF scheduler corresponds in the symmetric setting to a processor sharing queue with state-dependent service rates. Some flow-level analysis on the score-base scheduler can be found in [4]. The round robin, PF and maximum signal-to-noise ratio schedulers are also analyzed at the flow-level in [15] with a more detailed link level model.

Our analysis using a dynamic programming approach in the static setting is similar to the one in [14]. Finally, closest to the setting in this paper is the work in [9]. The authors introduce the TAOS2 scheduler, which gives the optimal one-step decision rule to improve the rate-oblivious standard SRPT scheduler. The authors have derived lower bounds on the performance of the scheduler in a static setting. Also, some experiments with the dynamic setting have been carried out. We include TAOS2 in our experiments, and also introduce a new set of priority-based schedulers, as well as schedulers using only knowledge of service attained thus far (non-anticipating schedulers). Additionally, we give results on the optimal policy in the static setting and an extensive set of simulations on the dynamic setting.

2. SIZE-BASED POLICIES VS. PF POLICY

Here we illustrate the difference between applying standard size-based schedulers that do not take advantage of the channel rate variations and the PF scheduler.

2.1 Assumptions

We consider downlink data transmissions in a cellular system, such as HSDPA, where the base station always transmits to one single user (or in our case a flow) within a time slot. The flows are experiencing a time varying channel, for example due to the fast fading phenomenon. The flows are independent from each other and the rate of each flow i varies over time according to some stationary process $R_i(t)$ and, as in, e.g., [6], it is assumed that the base station knows perfectly the rate $R_i(t)$ of each flow for each time slot. Furthermore, we assume that the channels of all the flows are symmetric, i.e., the stationary distribution of $R_i(t)$ is identical with all flows.

The traffic in the system consists of elastic flows corresponding roughly to file transfers that the users are downloading through the base station. The flows have a random size, denoted by X (bits), and typically require many time slots to finish their service. The flows arrive at the base station according to a Poisson process with intensity λ.

Finally, we consider the system in the limit when the time slot duration is negligible compared with the duration of the flows, see [5, 6, 1]. This allows us to use results on M/G/1 queues for assessing the performance impact of fast fading on standard size-based schedulers and the PF scheduler.

2.2 Different schedulers

So called standard size-based schedulers are not able to exploit the variation of the instantaneous rates of the flows. In the limit, when the time slot duration tends to zero, the rate variations average out and the flows are served according to the mean of the rate process. The system corresponds to an M/G/1 queue with arrival rate λ, service time $S = X/\mathrm{E}[R_i]$ and load $\rho = \lambda\,\mathrm{E}[S]$. Note that stability of the system requires $\rho < 1$. However, different schedulers can be derived depending on the accuracy of the flow-level size information.

PS policy: If the scheduler simply assigns time slots in a round robin manner, the service is approximated by the PS (Processor Sharing) discipline, which does not utilize size information at all.

FB policy: Within the stability region $\rho < 1$ it is possible to affect the mean delays of the flows by using knowledge of the flow sizes, see, e.g., [1]. The FB (Foreground Background) policy always serves the flow that has received least amount of service. Among non-anticipating policies, the FB policy minimizes the mean flow delay when service time distribution has the DHR property.

SRPT policy: The optimal scheduler minimizing the mean flow delay is given by SRPT, which always serves the flow with the smallest remaining processing time, implying exact knowledge of the remaining bits in the flow. However, the stability region for the SRPT policy (also for PS and FB) is still $\rho < 1$.

For the above standard size-based schedulers the equations for the conditional and total mean flow delays are readily available from the literature, see, e.g., [10]. Note that the results apply also in asymmetric scenarios. The asymmetric channels of the flows correspond to classes and just modify the service time distribution, see [1].

PF scheduler: The idea in opportunistic scheduling, on the other hand, is to exploit the rate variations between the different flows, such that time slots are allocated to the flow that happens to have good channel conditions in the considered time slot. A well-known example of channel-aware schedulers is the PF scheduler which allocates the time slot to the flow i that has the highest instantaneous rate R_i relative to its realized throughput. In the symmetric setting, the PF scheduler can be approximated at the flow-level by a processor sharing queue with state dependent service rates, see [6]. The PF scheduler benefits from multiuser diversity which increases the capacity of the system. In our considered symmetric modeling setting, this implies that the system can be stable also for $\rho > 1$. This is possible as ρ is defined based on the mean rate of the channel, and the PF scheduler exploits instantaneous rate variations, which essentially increases the capacity. However, the exact stability limit now depends on the channel properties. In this paper, we refer to the load region $\rho > 1$ as heavy traffic.

2.3 Numerical example

Next we illustrate the impact of rate variability on the relative performance of the size-based schedulers and the PF scheduler. Consider the following simple model, taken from [5], for the instantaneous rate of the flows, R,

$$R = \min\{c_0, m \cdot \xi\},$$

where c_0 represents the maximum achievable rate in the system, m represents the mean rate of the flow in the absence of fast fading and ξ is a random variable denoting the marginal distribution of the fast fading process. Thus, it is assumed that the instantaneous rate is linear in the received signal-to-noise-ratio up to a maximum rate c_0 with the parameter m modeling the impact of path loss and ξ the fast fading process. We assume that ξ is exponentially distributed (with mean 1), corresponding to a Rayleigh fading channel. Also, we choose the parameters such that $c_0 = 1$, the file sizes obey a Pareto distribution with shape parameter 2 (DHR type distribution) and the scale parameter b is determined so that the mean service time $\mathrm{E}[S] = 1$ (given the channel parameters m and c_0).

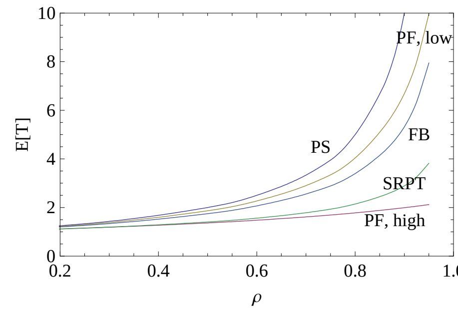

Figure 1: Comparison of the mean flow level delay as a function of ρ for PS, FB, SRPT and PF.

The parameter m controls the variability of the rates; low m means poor channel quality conditions (e.g., far from the base station) while a high m corresponds to good channel conditions (e.g., close to the base station). From the point of view of scheduling gain (PF scheduler), the best results are expected when m is low and correspondingly the rate variability is high (and thus scheduling gain is also high). As the parameters are chosen always so that $\mathrm{E}[S] = 1$, the performance of PS, FB and SRPT does not depend on m.

The results, based on known formulae, e.g., from [10, 6, 1], are shown in Figure 1, which displays the mean flow-level delay as a function of the load ρ for the different policies. For the PF policy two curves are given. The graph labeled "PF, low" corresponds to the low variability case with $m = 10$ and "PF, high" represents the high variability case with $m = 1$. It can be observed that when the rate variability is high, the scheduling gain enables the system to serve the flows at a much higher average rate and the resulting performance is much better than even under the optimal size-based scheduler SRPT. However, when the average channel condition is very good and scheduling gain becomes small, the size-based methods are clearly advantageous.

Our objective in this paper is to develop schedulers that utilize simultaneously both rate and size information in an attempt to further minimize the flow level delays. As seen from the previous results, it is very important to achieve a proper balance between the size and rate information depending on the channel conditions of the flows.

3. POLICIES COMBINING SIZE AND RATE INFORMATION

In this section we present the used policies for minimizing the flow-level delay. The policies are categorized depending on how the instantaneous rate information is utilized. Within each category there are variants of the policies based on the type of flow-level size information.

To define the policies, some notation is introduced. We consider the scheduling decision at time t. At time t there are $N(t)$ flows in the system and the instantaneous rate of each flow is denoted by $R_i(t)$ (as earlier). We assume that the possible values of $R_i(t)$ form a (finite) discrete set of size L, i.e., $R_i(t) \in \{r_1, \ldots, r_L\}$. For example, in HDR/HSDPA systems there are 11 possible rates. Furthermore, we assume that the rate process evolves at a fast time-scale so that the values of $R_i(t)$ in successive time slots constitute an independent and identically distributed (i.i.d.) sequence.

Associated with each flow i there may be knowledge about the size of the flow. The general idea in using this information is the same as in standard size-based scheduling, that is to favor small flows in an effort to reduce the number of ongoing flows in the system. We denote by $Y_i(t)$ the remaining number of bits to be served for flow i at time t. Use of $Y_i(t)$ forms the basis of SRPT-like policies. On the other hand, $A_i(t)$ denotes the attained service of flow i in bits at time t, i.e., the amount of bits that have been served from the flow by time t. Policies using $A_i(t)$ are FB-like policies. The time that flow i has spent in the system up to time t is denoted by $D_i(t)$. As the scheduling always concerns a given time slot t and all variables have instantaneous values at time t, we omit the explicit dependence on time from the notation.

3.1 Priority policies

In the priority policies, we give absolute priority to the flows that have currently the highest instantaneous rates. In this way, the utilization of the channel can be maximized. Given that there are N flows in the system, the set of instantaneous rates is $\{R_1, \ldots, R_N\}$ and we denote by $\mathcal{K}$ the set of flows having the highest rates in the current slot, i.e.,

$$\mathcal{K} = \{i : R_i = \max\{R_1, \ldots, R_N\}\}.$$

In case the maximum is not unique, depending on the accuracy of the flow-level size information, we have different policies for breaking the ties.

SRPT-P: If exact information about the remaining job sizes Y_i is available, we can use that to allocate the service to the flow with the highest instantaneous rate and least amount of remaining bits to be served. This gives us the SRPT-P (SRPT with Priorities) policy. Under this policy the scheduler always selects the flow i^* that achieves

$$i^* = \arg\min_{i \in \mathcal{K}} Y_i.$$

FB-P: If information about Y_i is not available, knowledge of attained service A_i can be used to approximate Y_i in the case that the flow size distribution has the DHR property. This gives us the FB-P (Foreground-Background with Priorities) policy, where the scheduler always selects the flow i^* that achieves

$$i^* = \arg\min_{i \in \mathcal{K}} A_i.$$

Fair-P: Information about the time that each flow has been in the system D_i and the attained service A_i can be used to increase the fairness between the flows. The ratio A_i/D_i represents the throughput of flow i and by favoring the flow with smallest throughput we arrive at the Fair-P (Fair with Priorities) policy. Under this policy, the scheduler chooses the flow i^* that achieves

$$i^* = \arg\min_{i \in \mathcal{K}} \frac{A_i}{D_i}.$$

In all above policies, in case i^* is not unique the decision is taken randomly.

3.2 Index policies

All policies here are index policies, where based on the channel information and the size information a distinct index is obtained for each flow in the system. The scheduler then chooses the flow with the highest index.

PF: The reference policy in our studies is provided by the PF scheduler, which aims to balance fairness and performance. The PF scheduler allocates the time slot to the flow i that has the highest instantaneous rate R_i relative to its throughput, which is typically estimated via an exponentially smoothed average of the realized rates, see [16, 3]. However, in our scenario the throughput of flow i is readily given by A_i/D_i. Therefore, we define the PF scheduler so that it selects the flow i that achieves

$$i^* = \arg\max_{i=1,\ldots,N} \frac{R_i}{A_i/D_i}.$$

RB: The relatively best (RB) scheduler, as defined in [2], on the other hand is a policy where the fairness is taken with respect to the expected quality of the channel. In more detail, the RB policy allocates the time slot to the flow that has the highest instantaneous rate relative to its mean rate, i.e., to the flow i^* achieving

$$i^* = \arg\max_{i=1,\ldots,N} \frac{R_i}{\mathrm{E}[R_i]}.$$

TAOS2: In [9], a policy has been devised, which is an SRPT-like policy exploiting knowledge of the remaining bits Y_i in each flow and the instantaneous rates R_i. The policy is based on considering a static setting with a fixed number of flows and represents a locally optimal scheduling decision that improves the standard SRPT, which only uses the mean rate of the flows. In more detail, let I_i denote the rank of flow i, when the flows are sorted in an ascending order according to $Y_i/\mathrm{E}[R_i]$ (i.e., the standard SRPT criterion). According to TAOS2, the scheduler selects the flow i^* that achieves

$$i^* = \arg\max_{i=1,\ldots,N} \left((N - I_i + 1) \frac{R_i}{\mathrm{E}[R_i]} \right). \quad (1)$$

As can be seen, TAOS2 is a product form rule with components related to rate and size information, and where the rate information is based exactly on the RB policy.

FB-TAOS2: In FB-TAOS2 we simply replace the knowledge of the remaining bits in flow i, Y_i, with knowledge of the attained service A_i. This gives us the non-anticipating version of TAOS2 and only affects the ranking of the flows. Thus, in FB-TAOS2 the flows are sorted in ascending order of $A_i/\mathrm{E}[R_i]$ and I_i denotes the rank of flow i and the scheduler selects the flow i^* according to (1).

Again, in all above policies, ties are decided randomly.

4. OPTIMAL POLICY IN STATIC SETTING

In this section we compare the performance of the SRPT-like policies (TAOS2 and SRPT-P) and the RB policy (with no size information) against the optimal policy that minimizes the mean delay. In general, constructing the optimal policy is very difficult. However, it can be done in a setting with 2 flows and 2 possible rates for the channel state.

4.1 Dynamic programming formulation of the optimal policy

With a suitable choice for the parameters, the evolution of the service of the 2 flows can be characterized by a discrete time Markov chain defined by the underlying rate process. Then the optimal solution can be obtained recursively via dynamic programming. Time evolves in a slotted manner and the time slot length is denoted by Δ. Let us denote the two possible rates of the channel by $r^{\min}$ and $r^{\max}$ and the probability that the rate in an arbitrary time slot equals $r^{\min}$ is denoted by p_1 for flow 1 and by p_2 for flow 2. All $\Delta, r^{\min}$ and $r^{\max}$ are assumed to be integer valued. The state of the chain is characterized by integers n_1 and n_2 denoting the amount of work left in flow 1 and 2, and r_1 and r_2 denoting the channel states of flow 1 and 2. Our objective is to minimize the total mean flow delay.

Given the state of the Markov chain (n_1, n_2, r_1, r_2), the decision is between serving flow 1 with rate r_1 or flow 2 with rate r_2. Independent of the decision, the incurred cost for the flow delay is equal to 2Δ (service during the current time slot delays both flows by Δ). The value function $v(n_1, n_2, r_1, r_2)$ gives the mean flow delay given the state and the following recursion determines the optimal policy to minimize the mean total completion time

$$\begin{aligned} v(n_1,&n_2, r_1, r_2) = 2\Delta + \\ &\min\{f((n_1 - \Delta r_1)^+, n_2), f(n_1, (n_2 - \Delta r_2)^+)\}, \quad (2) \end{aligned}$$

where the minimization is between giving the service to flow 1 with rate r_1 or flow 2 with rate r_2 and the function $f(\cdot,\cdot)$ determines the mean flow delay (or the cost) of the decision,

$$\begin{aligned} f(n_1, n_2) = &\, p_1 p_2 v(n_1, n_2, r^{\min}, r^{\min}) \\ &+ p_1(1 - p_2) v(n_1, n_2, r^{\min}, r^{\max}) \\ &+ p_2(1 - p_1) v(n_1, n_2, r^{\max}, r^{\min}) \\ &+ (1 - p_1)(1 - p_2) v(n_1, n_2, r^{\max}, r^{\max}). \end{aligned}$$

The initial value for the recursion is $v(0, 0, \cdot, \cdot) = 0$. If only the class 1 flow is left in the system the cost incurred given the state equals Δ and the recursion (2) takes the form

$$v(n_1, 0, r_1, r_2) = \Delta + f((n_1 - \Delta r_1)^+, 0),$$

and similarly in the case if $n_1 = 0$ but $n_2 > 0$. Note that, if a flow i finishes its service before the time slot ends (i.e., $n_i < \Delta r^{\min}$) the cost incurred is equal to Δ. Now, to compute the optimal policy and the corresponding minimized mean flow delay, one just uses the iteration formulae above and they are iterated until a given upper limit for the flow sizes.

4.2 Analysis of SRPT-P and TAOS2 policies

For the SRPT-P, TAOS2 and RB policies the mean flow delay can also be obtained. The idea is again to go through recursively the entire state space and given the state (n_1, n_2, r_1, r_2) the minimization operation in (2) is replaced by the decision rule of the corresponding policy.

By examining in detail the SRPT-P and TAOS2 policies, it is also easy to show that they are very similar in the case of two possible rates, two jobs and symmetric channels, i.e., $p_1 = p_2$. In fact, the only differences arise with two possible states. Consider the TAOS2 scheduler. If $I_1 = 1, I_2 = 2, r_1 = r^{\min}$ and $r_2 = r^{\max}$, then the index values of the flows are $\{2r^{\min}, r^{\max}\}$. Thus, the TAOS2 scheduler will select flow 2 only if $r^{\max} \geq 2r^{\min}$, and otherwise flow 1. On the other hand, under SRPT-P the scheduler would always select flow 2 based on the priority of the higher rate for flow 2. A similar phenomenon occurs also if the state is $I_1 = 2, I_2 = 1, r_1 = r^{\max}$ and $r_2 = r^{\min}$. Thus, the SRPT-P and TAOS2 policies are in the symmetric 2 flows/2 rates setting equal if $r^{\max} \geq 2r^{\min}$. Note that this does not hold anymore in the same asymmetric setting nor when there are more than two rates in the system or more jobs.

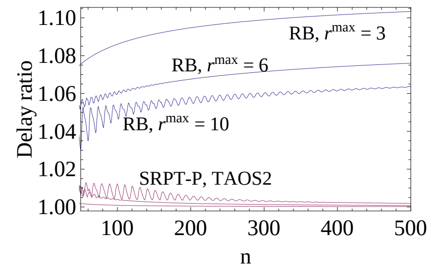

Figure 2: Ratio of the mean flow delay under RB and both TAOS2 and SRPT-P to the optimal scheduler performance for a diagonal cross section (n, n).

4.3 Numerical results

Here we compare the mean flow delay under the various heuristics (RB, TAOS2, SRPT-P) against the optimal policy. For convenience, we select $\Delta = 1$ and $r^{\min} = 1$ as fixed parameters. In all cases, the results are shown only for the initial channel state $(r^{\min}, r^{\min})$, but the results are practically the same also for other initial channel state.

We first consider a symmetric setting $p_1 = p_2 = 0.5$ for different values of the higher rate $r^{\max} = \{3, 6, 10\}$ (SRPT-P and TAOS2 are identical in all cases). We study a cross section of the state space, where both flows have equal size and we evaluate the relative performance against the optimal as a function of the common size $n = \{1, \ldots, 500\}$. For the RB policy, the $r^{\max}$ values are indicated in the figure. For the SRPT and TAOS2 policies this is not possible, but the lines are ordered so that $r^{\max} = 10$ corresponds to the topmost curve and $r^{\max} = 3$ to the lowest curve.

The results are given in Figure 2, where the oscillations especially for small n are due to the discrete nature of the system. For large n the discretization effects get smaller and the graphs become smoother. The gain obtained with the optimal policy over the RB policy is noticeable and it increases with the common size, suggesting a persistent benefit from using size information as the size of the flows increases. On the other hand, as $r^{\max}$ is increased from 3 to 10 the benefit decreases, i.e., under higher channel variability RB becomes closer to the optimal policy. On the other hand, for the SRPT-P and TAOS2 schedulers the benefit from the optimal policy is very small and it also decreases as the common size increases, suggesting that the optimal policy differs from the TAOS2 and SRPT-P policies only when the flow sizes are small. Also, increasing $r^{\max}$ only marginally improves the performance for smaller flow sizes.

To see the impact of the asymmetric setting, we consider a case where for class 1 we set $p_1 = 0.8$ and for class 2 we set $p_2 = 0.2$. Thus, the overall probability to have rate $r^{\min}$

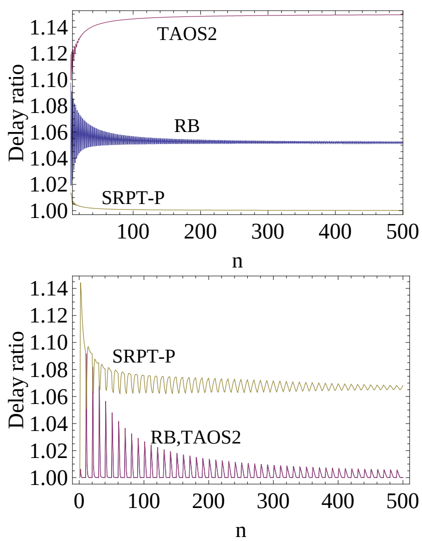

Figure 3: Ratio of the mean flow delay under RB, TAOS2 and SRPT-P to the optimal scheduler performance for $r^{\max} = 2$ (upper) and $r^{\max} = 10$ (lower) for a diagonal cross section (n, n).

is 0.5, i.e., the same as in the symmetric setting. First we consider the case with a relatively small rate variability and set $r^{\max} = 2$ and the results are shown in Figure 3 (upper), which depicts the ratio of the mean flow delay under the RB, TAOS2 and SRPT-P policies to the optimal scheduler performance for a diagonal cross section of the state space, (n, n) where $n = \{1, \ldots, 500\}$. The results show that as the amount of work in the system increases, the priority policy SRPT-P is close to optimal, the RB policy is somewhat poorer but the TAOS2 policy performs worst. Next we consider the case with a higher rate variability where $r^{\max} = 10$. The results are in Figure 3 (lower), which shows the results for the same diagonal cross section (n, n). In this case, RB and TAOS2 are very close to the optimal and somewhat surprisingly the priority policy SRPT-P is not as efficient. This can be due to the fact that in this highly asymmetric setting, the RB and TAOS2 policies are able to separate the two classes from each other and favor class 1 (class 1 has the bad channel) when it happens to have a high rate, while the SRPT-P policy treats both classes equally in this respect.

5. HEAVY TRAFFIC SIMULATIONS

In this section we consider the performance of the various schedulers introduced in Section 3 in a dynamic setting where flows arrive according to a Poisson process and have random sizes. The idea is to study the heavy traffic behavior of the policies, i.e., with $\rho > 1$, under different settings for the flow rates (2 possible rates vs. 11 possible rates) both in a symmetric and asymmetric scenario.

The simulations are carried out using two flow classes both with equal Poisson arrival rates of flows, $\lambda_1 = \lambda_2 = 0.5$, i.e., the total arrival rate equals 1. The file sizes obey the Pareto distribution with shape parameter $\beta = 2$. This renders the variance of X infinite, which is a property that has been observed in measurements of elastic traffic, see [7]. To have a given load in the system, we vary the mean file size $\mathrm{E}[X]$ by choosing the remaining parameter of the Pareto distribution, i.e., the scale parameter b. By using two flow classes we can analyze both symmetric and asymmetric channel configurations by appropriately parameterizing the channel rate processes.

In the experiments with only 2 rates, the rates of the two flow classes are characterized by p_1 and p_2 denoting the probability that the channel rate is $r^{\min}$ for flow classes 1 and 2, respectively. We assume that $p_2 \le p_1 < 1$, i.e., that class 2 has an equally good or better channel than class 1. For this setting it is possible to select p_1 and p_2 such that the rate processes appear statistically identical for an outside observer both in the symmetric and asymmetric case.

In the case of 11 different rates as in HSDPA/HDR systems, $R_i \in \{r_1, \ldots, r_{11}\}$, for classes $i = 1, 2$. For the absolute values of the rates, see, e.g., [6]. We assume that the probabilities of the different rates obey a truncated geometric distribution with parameter $q \le 1$ giving us an easy way to parameterize the rate distribution (only 1 parameter is required). For class 1 the rates then obey the truncated geometric distribution with parameter q, while for class 2 the rate probabilities are a mirrored version of class 1, i.e.,

$$\mathrm{P}\{R_1 = r_k\} = \mathrm{P}\{R_2 = r_{11-k+1}\} = \frac{q^k}{\sum_{i=1}^{11} q^i}.$$

In the simulations, the idea is to investigate the performance of the policies as a function of the total load. To this end we define the load ρ as in Section 2, i.e., it is the load in the corresponding system where instantaneous channel variations are not used by the scheduler and the service rate of the system is simply defined by the mean rate of the channel. Thus, the load ρ is given by

$$\rho = \lambda_1 \frac{\mathrm{E}[X]}{\mathrm{E}[R_1]} + \lambda_2 \frac{\mathrm{E}[X]}{\mathrm{E}[R_2]}, \tag{3}$$

where $\mathrm{E}[R_1]$ and $\mathrm{E}[R_2]$ denote the mean rates of class 1 and 2, respectively. Note that, in the corresponding M/G/1 queue, stability requires that $\rho < 1$.

With the opportunistic schedulers proposed in Section 3, it is possible to increase load beyond the region $\rho < 1$, but it is not known exactly how much and one objective in the simulations is to experimentally determine roughly the stability limits. However, an upper bound on the stability limit is obtained by assuming all flows in both classes are served at the common maximum rate, $r^{\max}$. Then, for the system to be stable, it must hold that $\lambda_1 \mathrm{E}[X]/r^{\max} + \lambda_2 \mathrm{E}[X]/r^{\max} < 1$. Solving this for equality (stability limit) yields $\mathrm{E}[X] = r^{\max}/(\lambda_1 + \lambda_2) = r^{\max}$. Thus, an upper bound ρ^* on the stability limit is given by (3) with $\mathrm{E}[X] = r^{\max}$,

$$\rho^* = \lambda_1 \frac{r^{\max}}{\mathrm{E}[R_1]} + \lambda_2 \frac{r^{\max}}{\mathrm{E}[R_2]}. \tag{4}$$

Essentially, ρ^* as given by (4) captures how much greater the stability limit under the assumption of maximum rate service is relative to the M/G/1 stability limit $\rho = 1$. The tightness of (4) depends on the channel rate parameters.

Finally, the simulations are conducted in the setting where the time slot length Δ is very small compared with time scale of arrivals and departures. In our simulations, we set $\Delta = 0.01$ time units.

Recall that the PF, RB, FB-TAOS2, TAOS2 policies are index policies, and SRPT-P, FB-P and Fair-P are priority policies. Additionally, the policies use different types of age information: SRPT-P and TAOS2 apply exact knowledge

Table 1: **Classification of the different policies.**

	Fair	FB	SRPT
Index	PF, RB	FB-TAOS2	TAOS2
Priority	Fair-P	FB-P	SRPT-P

of sizes (SRPT-like criterion), FB-P and FB-TAOS2 apply knowledge of service attained thus far (FB-like information), and RB, PF and Fair-P aim at some kind of fairness (RB does not explicitly aim at fairness while PF and Fair-P do). The different policies can be classified according to the priority/index and size information usage as shown in Table 1.

5.1 Symmetric scenario

In the symmetric setting we have $p_1 = p_2 = 0.5$ so that all flows have statistically identical channels. The performance of the system is represented by the mean number of flows in the system, E$[N]$, which in our case is also equal to the mean delay E$[T]$ since $\lambda_1 + \lambda_2 = 1$. For each value of ρ, 10 repeated runs were made with 10^6 flows in each run.

In the first experiment we have only two possible rates in the system and we keep $r^{\min} = 1$. Figure 4 depicts the mean number of flows for the different schedulers as a function of the load ρ when $r^{\max} = 2$ (upper) and $r^{\max} = 20$ (lower), respectively. In the figure, the index policies are shown with dashed lines while the priority policies are indicated with solid lines. In the figure, the labels Fair, FB and SRPT refer to the classification in Table 1. The fair index policy corresponds to PF. The RB policy is not shown separately as it is in the symmetric setting very similar to PF. The value of ρ^* in both cases is shown as a vertical line.

More specifically, from the upper panel (low rate variability) it can be seen that all other policies perform better than PF (index policy, fair). Also, the SRPT- and FB-like policies separate nicely so that the SRPT-like policies are consistently better than FB-like policies. Finally, the corresponding priority based scheme is always better than its TAOS-variant. In the lower panel, we have $r^{\max} = 20$ and hence the rate variability is higher. In this setting, we clearly see that the priority policies (FB-P, SRPT-P) become practically identical with the corresponding index policies (FB-TAOS2, TAOS2). At a very high load, there is some difference, but also the confidence intervals for the last points are wider. Again, all policies yield better results than PF (index policy, fair). Note also that the capacity limit in lower panel is higher due to a higher scheduling gain. In both figures, the approximate capacity limit ρ^* seems to be quite tight.

Finally we look at the case with 11 possible rates as in the HSDPA/HDR systems. To have a symmetric setting for the two flow classes, we use $q = 1$ as the parameter in the truncated geometric distribution of class 1 yielding a uniform distribution for the rates in both classes (recall that for class 2 the rate probabilities are reversed from class 1). The simulation runs consisted of 10 repeated simulations with 10^6 flows in each run. Figure 5 gives the results again as a function of ρ, where the index policies are shown with dashed lines and the priority policies are depicted with solid lines. Now the index and priority policies separate into different groups. Surprisingly, both TAOS2 and FB-TAOS2 policies are worse than PF. However, the priority based policies are all performing slightly better than PF and their relative performance is ordered as expected (SRPT-like policy is best, FB-like policy next, and the fair variant Fair-P is last). The capacity limit is also higher than earlier due to a higher scheduling gain (e.g., the ratio $r^{\max}/r^{\min} \approx 60$). However, the approximate capacity limit is in this case $\rho^* \approx 3.4$ (not shown in the figure), and it does not seem particularly tight.

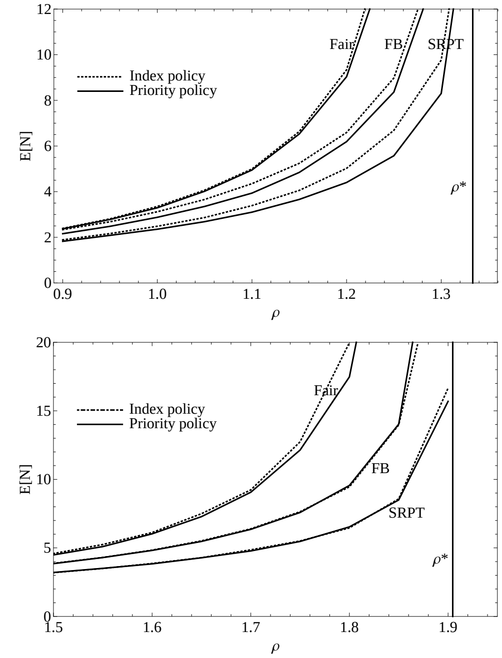

Figure 4: The mean number of flows (symmetric setting, 2 rates) for the different schedulers as a function of the load ρ when $r^{\max} = 2$ (upper) and $r^{\max} = 20$ (lower), respectively.

5.2 Asymmetric scenario

Next we study the asymmetric setting where the flows in class 1 have different channel properties than the flows in class 2. First we consider the case with only 2 possible rates. The parameters are chosen so that $p_1 = 0.8$ and $p_2 = 0.2$, i.e., for class 1 the probability of having rate $r^{\min}$ is 0.8 while

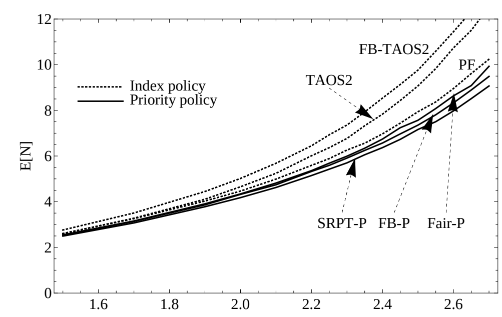

Figure 5: The mean number of flows (symmetric setting, 11 rates) for the different schedulers as a function of the load ρ in a system with the same rates as in HSDPA/HDR systems.

for class 2 the same probability equals 0.2. Thus, class 1 has worse channel conditions than class 2. Note that with the above values for the rate probabilities, the overall probability to have rate $r^{\min}$ is still $0.5p_1 + 0.5p_2 = 0.5$, i.e., exactly the same as in our corresponding symmetric scenario. Hence, the rate process appears identical to an outside observer in the corresponding symmetric and asymmetric scenarios. The simulations consisted of 10 repeated simulations with 10^6 flows in each run.

For the performance we study first the mean number of flows in the system allowing us to see how the schedulers perform overall. Figure 6 shows the results as a function of the load ρ for $r^{\max} = 2$ (upper) and $r^{\max} = 20$ (lower). In the figure, the index policies are shown with dashed lines and the priority policies with continuous lines. The labels Fair, FB and SRPT correspond to the classification in Table 1. The fair index policy again corresponds to the PF scheduler. However, in the asymmetric settings the RB policy displays a quite different behavior than PF and hence it is indicated separately. The approximate capacity limit ρ^* is again displayed as a vertical line.

In both panels of Figure 6, at higher loads, considering first only the priority policies, the policies are sorted according to the size information (SRPT-P is best, FB-P next and Fair-P last). Among the index policies TAOS2 is best, PF is practically the same as FB-TAOS2 in the upper panel while in the lower panel FB-TAOS2 is better than PF, and the RB policy is worst. However, the FB- and SRPT-like policies do not separate nicely anymore as in the symmetric setting and the relative performance of the policies is more complex. Comparing the policies to the PF policy (index, fair), all policies using size information, except FB-TAOS2 in the low variability case (upper panel), perform better than PF. The RB scheduler performs clearly the worst. The priority based policies SRPT-P and FB-P typically perform better than the corresponding TAOS2 policy. However, in the high variability case (lower panel) somewhat surprisingly it is the TAOS2 policy that shows better performance than SRPT-P (thus far the SRPT-P policy has always been better). This can be partly explained by the behavior already seen in the static asymmetric setting in Figure 3 (lower). Again, the approximate capacity limit ρ^* appears to be rather good for the TAOS2 and SRPT-P policies.

Comparing the symmetric and asymmetric settings with respect to the capacity regions (see Figures 4 and 6), it can be observed that the capacity limit in the asymmetric setting is higher than in the symmetric setting with respect to the capacity limit of the rate-oblivious M/G/1 system, i.e., the load ρ can be extended farther beyond 1 in the asymmetric setting than in the symmetric setting. Recall that the rate processes in the symmetric and asymmetric setting are parameterized so that in both cases for an outside observer the probability of having rate $r^{\min}$ or $r^{\max}$ equals 0.5. The seemingly larger stability region can be partly explained by the fact that in the asymmetric setting, the scheduler can separate the asymmetric classes from each other to make more efficient use of the time slots.

Finally, we consider the case with 11 different rates as in the HSPDA/HDR systems. The results on the mean number of flows in the system as a function of the load ρ are shown in Figure 7, where the upper panel corresponds to a scenario with a relatively low degree of asymmetry between the flow classes ($q = 0.9$) and in lower panel the asymmetry

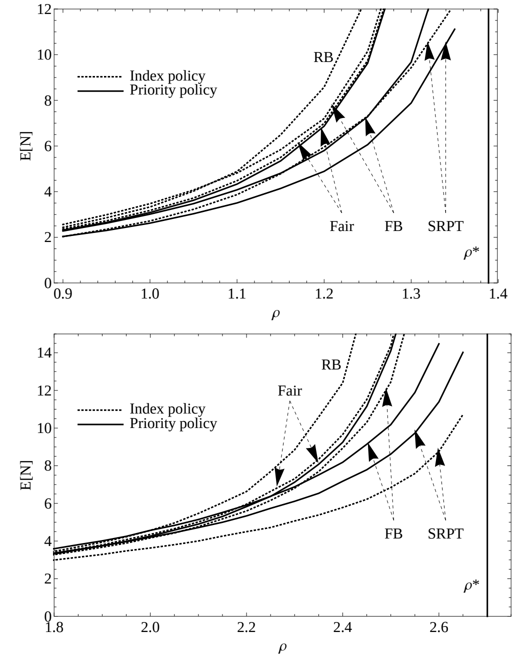

Figure 6: The mean number of flows (asymmetric setting, 2 rates) as a function of the load ρ with $r^{\max} = 2$ (upper) and with $r^{\max} = 20$ (lower).

is quite high ($q = 0.5$). The simulations consisted of 10 repeated simulations of 10^5 flows. The upper panel displays similar characteristics as the symmetric scenario in Figure 5. The TAOS2 index policies and the RB policy are performing worse than PF, while the priority policies have a slightly better performance than PF. The relative order of the priority policies at the highest load is also as in Figure 5. However, the confidence intervals at the highest load still overlap, so a definite conclusion can not be drawn. When the degree of asymmetry increases (see the lower panel), the situation changes and all index and priority policies perform in a very similar manner up to a rather high load. The priority policies are marginally better than the index policies FB-TAOS2 and TAOS2. At very high load the RB policy first becomes worse and then the FB-TAOS2 policy. Notably, the PF policy performs the best throughout the whole load region. In these cases, the approximate capacity limits are $\rho^* \approx 3.7$ for $q = 0.9$ and $\rho^* \approx 14.9$ for $q = 0.5$, and they do not seem to be particularly accurate, especially for $q = 0.5$.

5.3 Fairness in asymmetric scenarios

Finally, we investigate the fairness properties of the schedulers in the asymmetric setting. The results corresponding to the same scenarios as in Figure 6 are shown in Figure 8. In the scenario of the upper panel, the mean rate of class 1 (bad channel) is 1.2 and for class 2 (good channel) it is 1.8, and thus the rate asymmetry between the classes is 1.5. In the scenario of the lower panel, the rate asymmetry is higher equalling approximately 3.4. The fairness measure corresponds to the metric, $\max\{E[T_1], E[T_2]\}/\min\{E[T_1], E[T_2]\}$, where $E[T_1]$ and $E[T_2]$ denote the mean delay of class 1 and 2, respectively. Essentially, this metric represents how far the performance of the two classes are from each other. The

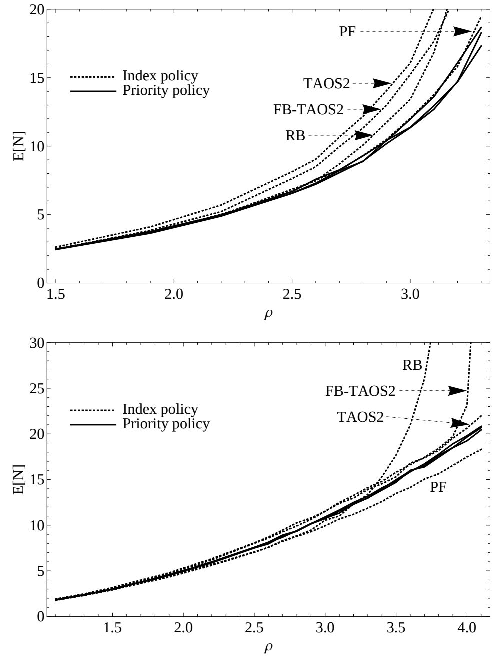

Figure 7: The mean number of flows (asymmetric setting, 11 rates) for the different schedulers as a function of the load ρ in a system with the same rates as in HSDPA/HDR systems with low asymmetry (upper) and high asymmetry (lower).

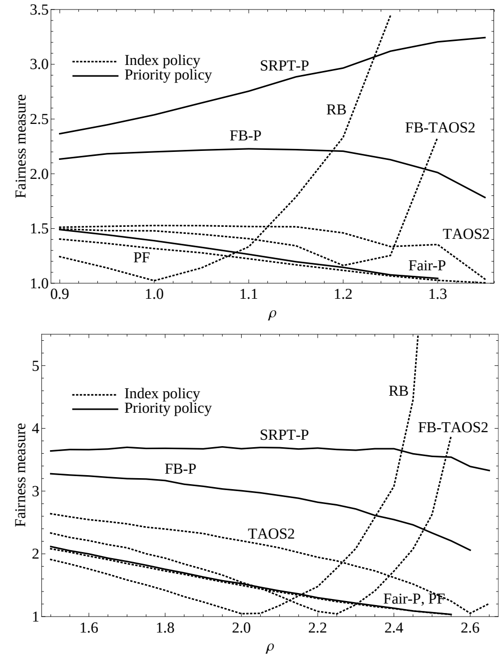

Figure 8: The fairness measure between the two classes in a system with 2 rates as a function of the load ρ for $r^{\max} = 2$ (upper) and $r^{\max} = 20$ (lower).

results reveal the non-monotonous behavior of the fairness for the RB and FB-TAOS2 policies (and the same behavior presumably holds also for TAOS2 if load would be further increased), which all use the same relative metric for giving a weight to the instantaneous rate. The reason for the non-monotonicity is that in the beginning class 1 flows have a higher delay (recall that that class 1 flows have a bad channel), but as the load is increased, which also increases the number of class 1 flows, the situation eventually changes: the scheduler will more often find a flow in class 1 having a sufficiently high rate which gives a better index value than for any class 2 flow. At this point the flow delay in class 2 starts increasing rapidly. On the other hand, the priority policies behave in a quite stable manner with the SRPT-P policy being somewhat more unfair than the FB-P policy. Notably the priority based fair policy Fair-P and the PF policy both achieve a good balance between performance and fairness so that at high loads fairness measure tends to 1.

Figure 9 shows the fairness properties of the schedulers in the same scenario with 11 rates as in Figure 7. The upper panel gives the results for the low asymmetry case and the lower panel for the high asymmetry case. In the low asymmetry case the fairness behavior is similar to that already seen in Figure 8, i.e., the index policies RB, FB-TAOS2, TAOS2 demonstrate the non-monotonous behavior, while the priority policies SRPT-P and FB-P are becoming increasingly less fair as load increases. The fair priority policy, Fair-P, and PF give a good balance between performance and fairness. However, in the high asymmetry case, all priority policies are quite unfair, even the Fair-P policy, while PF behaves more stable with respect to fairness.

6. CONCLUSIONS

In this paper we have considered the optimal scheduling problem of downlink time slots in an HSDPA/HDR system to minimize the mean flow delay of elastic traffic by utilizing simultaneously both instantaneous rate information and information about the flow sizes. We have assumed that at the flow-level the channel offers a discrete set of possible rates, the flows are independent from each other and that the channel rate varies independently in each time slot.

Several schedulers have been derived which differ in the way that the rate information is used. In the priority based policies, absolute priority is always given to the flows with highest instantaneous rates and, within this set of flows, size based information is applied. In the other class of policies, we have index policies which utilize a relative metric for the rates combined with size dependent information. The size information is related to knowing exactly the size of the flow or only knowing the attained service thus far.

In a static setting with two flows and two possible channel rates, the optimal policy can be constructed via a dynamic programming approach and the policies using exact size knowledge can be compared against this. The results showed that in the symmetric setting both TAOS2 and the SRPT-P policies were close to optimal. However, in the asymmetric setting the relative performance of the TAOS2 and SRPT-P policies depended heavily on the parameters.

To analyze the performance of the schedulers in the dynamic setting with stochastically arriving flows, several simulation experiments were made under heavy traffic conditions. Compared with the standard PF scheduler the results showed that in the extreme case with only 2 possible rates, clear performance improvements can be obtained with combined use of rate and size information. However,

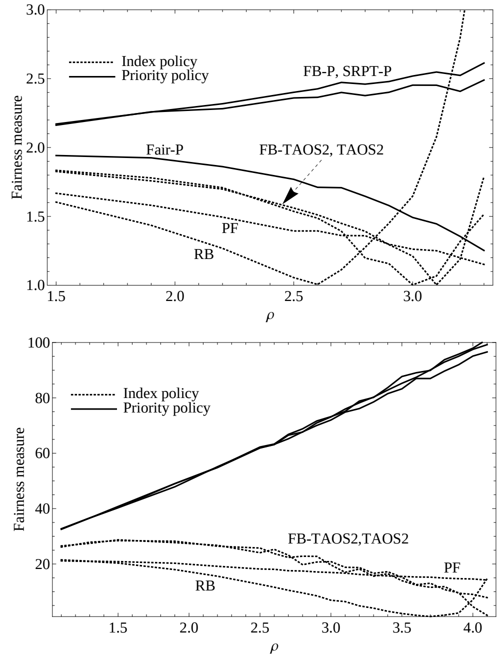

Figure 9: The fairness measure between the two classes in a system with 11 rates (HSDPA/HDR case) with low asymmetry (upper) and high asymmetry (lower).

when the number of possible rates is increased the benefits to be gained decrease substantially. Additionally, in the symmetric setting the priority based policies offer better performance than the TAOS2-based policies. Also, the difference between using exact knowledge of the flow sizes or only knowing the service attained thus far could be clearly separated. In the asymmetric setting, the differences were not as easily discernible. For example, depending on the scenario, TAOS2 can be better than the priority based SRPT-P. In the fairness behavior, the index policies, except PF, exhibit a non-monotonous behavior, while the priority policies have a more stable behavior (if asymmetry is moderate).

Future research contains still many more open issues. For example, even in the considered simple channel model setting the optimal scheduler is not known, and more research is required for its characterization. Also, the stability properties of the studied heuristics require more analysis. Of more practical interest can be to consider the impact of imprecise feedback information. Also, the same scheduling problem could be studied under slowly varying channels reflecting the fact that the state of the channel in successive time slots may not be independent.

7. REFERENCES

[1] S. Aalto and P. Lassila. Impact of size-based scheduling on flow-level performance in wireless downlink data channels. In *Proceedings of the* 20^{th} *International Teletraffic Congress*, pages 1096–1107, June 2007.

[2] F. Berggren and R. Jäntti. Asymptotically fair transmission scheduling over fading channels. *IEEE Transactions on Wireless Communications*, 3(1):326–336, Jan. 2004.

[3] T. Bonald. A score-based opportunistic scheduler for fading radio channels. In *Proceedings of European Wireless*, pages 283–292, Feb. 2004.

[4] T. Bonald. Flow-level performance analysis of some opportunistic scheduling algorithms. *European Transactions on Telecommunications*, 16(1):65–75, 2005.

[5] T. Bonald and A. Proutière. Wireless downlink data channels: user performance and cell dimensioning. In *Proceedings of ACM MobiCom*, pages 339–352, Sept. 2003.

[6] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking*, 13(3):636–647, 2005.

[7] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. In *Proceedings of ACM SIGMETRICS*, pages 160–169, May 1996.

[8] J. Holtzman. Asymptotic analysis of proportional fair algorithm. In *Proceedings of IEEE PIMRC*, volume 2, pages 33—37, Sept. 2001.

[9] M. Hu, J. Zhang, and J. Sadowsky. Traffic aided opportunistic scheduling for wireless networks: algorithms and performance bounds. *Computer Networks*, 46(4):505–518, 2004.

[10] L. Kleinrock. *Queueing systems, vol II: computer applications.* John Wiley & Sons, first edition, 1976.

[11] S. Patil and G. de Veciana. Measurement-based opportunistic scheduling for heterogenous wireless systems. submitted to IEEE Transactions On Communications, Jan. 2006.

[12] R. Righter and J. Shanthikumar. Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures. *Probability in the Engineering and Informational Sciences*, 3:323–333, 1989.

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

[14] B. Tsybakov. File transmission over wireless fast fading downlink. *IEEE Transactions on Information Theory*, 48(8):2323–2337, Aug. 2002.

[15] H. van den Berg, R. Litjens, and J. Laverman. HSDPA flow level performance: the impact of key system and traffic aspects. In *Proceedings of ACM MSWiM*, pages 283–292, Oct. 2004.

[16] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48(6):1277–1294, jun 2002.

[17] S. Yashkov. Processor-sharing queues: Some progress in analysis. *Queueing Systems*, 2:1–17, 1987.