Flow-level stability and performance of channel-aware priority-based schedulers

Samuli Aalto, Pasi Lassila
Aalto University School of Science and Technology, Finland
Email: firstname.lastname@tkk.fi

***Abstract*—Channel-aware scheduling in modern wireless networks enables the system to exploit the random rate variations across different users to increase the performance of the system. We analyze channel-aware priority-based downlink scheduling policies at the so-called flow level with a stochastically varying number of users. The priority can be any monotonously increasing function of the instantaneous rate of the user, which generalizes the well-known linear weight-based policies. Also, ties are allowed within a user class, as well as between user classes. As the main result, we characterize when these priority-based policies are stable under an intuitive necessary condition, which holds for arbitrary tie breaking rules and is independent of the flow size distribution. Additionally, for the policies for which the necessary condition is not sufficient, a more stringent condition is derived in the case of two traffic classes. Finally, extensive simulations have been performed to compare the performance of different priority-based and utility-based policies.**

I. Introduction

One key aspect in modern cellular systems affecting the performance is the scheduling of the radio resources among the users. In these systems, time is slotted and scheduling decisions can be made at the time scale of milliseconds. The fast scheduling and channel quality feedback enables the system to optimize the scheduling for data traffic by exploiting the fading phenomena causing random variations in the channel quality across the users. These channel-aware (also known as opportunistic) schedulers aim at increasing the throughput of the system by favoring those users having instantaneously good channels. A well-known example is the proportionally fair (PF) scheduler proposed for 1xEV-DO systems, see [1].

Channel-aware schedulers can be broadly classified as utility/maxweight based and rate-based priority schedulers. Utility/maxweight based approaches use instantaneous rate information coupled with knowledge of the throughput/queue sizes/packet delays, see [2], [3], [4]. The rate-based approaches, such as the relatively best (RB) scheduler [5] and its generalizations to weight-based strategies [6], only use information about the rate (channel) statistics. Typically, these channel-aware schedulers have been analyzed at the time slot or the packet level with a static population of users, either assuming a saturated set of users, see [5], [4] or allowing packet level-dynamics and analyzing the system at the heavy traffic limit [2], [7]. The results have established that simple stochastic gradient policies of the utility and so-called maxweight policies (of either packet delays or queue lengths) have many desirable properties. However, in the above models the common aspect is that the number of users stays constant, which is not entirely valid when considering the performance of elastic data traffic.

We consider channel-aware downlink scheduling in a dynamic setting at the so-called flow level assuming that the scheduler at the base station always transmits to exactly one user at a time. Flows correspond roughly to file transfers that the users are downloading through the base station. Flows arrive stochastically and have random sizes. The flow-level abstraction implies that the scheduler observes a randomly varying achievable rate for each flow in every time slot, and the system is considered in the limit where the time slot duration goes to zero. The set of achievable rates is discrete, but different user classes may have a different stationary distribution for the rates. Performance at the flow level is expressed, *e.g.*, as the throughput or the mean flow delay.

In this paper we focus on the flow-level performance and stability properties of certain channel-aware scheduling policies. Stability relates, *e.g.*, to the situation where flows are grouped into classes with different stationary distributions for the achievable rates (cf., good channel/bad channel) and stability implies that the number of flows (and the flow delays) stays finite. While stability in practical systems is guaranteed by some form of admission control and user impatience, it is still an important indicator of the robustness of the policy under unpredictably varying load conditions.

The necessary condition for stability essentially states that at the stability limit the scheduler always serves the flows according to the class-specific maximum achievable rates. Important work in this area has already been done in [6], [8], [9]. In [6] it has been shown that for the weight-based priority policies the necessary condition may not be sufficient, and the sufficient conditions have been characterized. Also, in [8], [9] it has been shown that for the utility-based policies of the per user throughput (including PF and other α-fair policies [10]), the necessary condition is sufficient.

The main result of our paper is the characterization of the prerequisites under which the necessary condition is sufficient for a general class of rate-based priority policies. The priority of a flow can be any monotonously increasing function of its instantaneous rate. The rate-based priority policies include as special cases the weight-based strategies in [6] representing the case of a linear priority function, and the schedulers based on the cumulative distribution function of the instantaneous rate [11], which corresponds to a non-linear priority function. Unlike in [6], we allow the same priority to appear between the flows of different classes, which is an essential extension when the transmission rates take values in a finite set. Moreover, we do not restrict ourselves to the randomized tie-breaking rule (as in [6]) but allow any rule within priority classes. The result holds for arbitrary flow size distributions. For the rate-based priority policies for which the necessary condition is not sufficient, a more stringent condition is derived that guarantees stability in the case of two traffic classes.

In addition to the stability analysis, we evaluate the performance of different policies by means of simulations. We experiment with a number of rate-based priority policies, *e.g.*, the RB policy, as well as with utility-based policies, namely PF and the α-fair policy with $\alpha = 2$ (minimizing the so-called potential delay [12]). The results show the potential instability problems of certain priority policies. Also, while some of the schedulers may be stable, the resulting performance can be rather poor. We then experiment with ideas to minimize the flow-level delay. Assuming that the base station does not use instantaneous rate information, the system can be modeled as an M/G/1 queue and the benefits of size-based scheduling prove to be significant, see [13]. However, combining size-based and opportunistic scheduling is difficult and no strong structural results on the optimal policy exist. Our experiments propose that if the scheduler can use instantaneous rate information, the size-based information has a minor role. Instead, the flow level approach can be used for the optimal choice of a channel-aware scheduler, *e.g.*, among the α-fair schedulers.

The paper is organized as follows. In Section II we define our model, while the used policies are introduced in Section III. Section IV gives the stability results of the paper. Numerical results and simulation studies are presented in Section V, and conclusions are given in Section VI.

II. Model and preliminaries

We consider downlink data transmissions in a cellular system, where the base station always transmits to one single user within a time slot, as in 1xEV-DO systems. The traffic consists of elastic flows (corresponding roughly to file transfers) that the users are downloading through the base station. The flow sizes, X_i (in bits), are independently and identically distributed with mean $\bar{x} = E[X_i]$. In the rest of the paper we refer to flows and users interchangeably.

The achievable rate (in bps) of a flow with index i in a time slot t varies over time according to some stationary and ergodic process $R_i(t)$ taking values in a finite set $\mathcal{R} = \{r_1, \ldots, r_J\}$ such that $r_1 < \ldots < r_J$. The finiteness and discrete nature of the set is motivated by the fact that in the practical systems, the achievable rates are defined by the adaptive coding and modulation employed in the system which render the set of achievable rates discrete and finite. In addition, we assume that the rate processes are independent from each other.

We assume that there are K different *user classes*. Within each class k, the rate processes are assumed to be identically distributed. Denote

$$F_k(r) = P\{R_k \leq r\}, \quad r > 0,$$

where R_k denotes a generic random variable with as distribution the stationary rate distribution of class k. Let $\bar{r}_k = E[R_k]$. Also, let $\mathcal{R}_k$ denote the set of possible class-k rate values,

$$\mathcal{R}_k = \{r \in \mathcal{R} : P\{R_k = r\} > 0\}.$$

Without loss of generality, we assume that $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_K$. Denote the highest rate related to user class k by

$$r_k^* = \max \mathcal{R}_k,$$

and the second highest rate by

$$r_k^{**} = \max(\mathcal{R}_k \setminus \{r_k^*\}).$$

All these statistical properties of various rate processes are assumed to be known, e.g., via measurements, for the base station when making the scheduling decisions.

The different user classes can model, *e.g.*, the fact that users may have channels with a different quality. In a practical system, if the mean SNR (signal-to-noise-ratio) is very low it can render the probabilities of the highest rates astronomically small. This situation can be modeled by setting the probabilities of such rates equal to zero for the corresponding class.

We consider a dynamic setting where the set of flows, $\mathcal{N}(t)$, varies randomly over time t. Let $\mathcal{N}_k(t)$ denote the subset consisting of class-k flows in the system at time t. We assume that new flows of class k arrive according to an independent Poisson process with rate λ_k. The total arrival rate is denoted by $\lambda = \lambda_1 + \ldots + \lambda_K$. The traffic load of class k (with respect to its maximum rate) is denoted by ρ_k^*,

$$\rho_k^* = \frac{\lambda_k \bar{x}}{r_k^*},$$

and the total load by $\rho^* = \rho_1^* + \ldots + \rho_K^*$. A flow departs from the system as soon as all its X_i bits have been transmitted. The amount $Y_i(t)$ of remaining bits of flow i decreases with rate $R_i(t)$ if flow i is scheduled for time slot t.

We assume that the base station perfectly knows the rate $R_i(t)$ of each flow $i \in \mathcal{N}(t)$ in the beginning of each time slot t when making scheduling decisions. Based on these rates, exactly one of these flows, $I(t) \in \mathcal{N}(t)$, is scheduled for time slot t (assumed that $\mathcal{N}(t) \neq \emptyset$). If $\mathcal{N}(t) = \emptyset$, then we define $I(t) = 0$. Note that the scheduling policies we consider are *work-conserving*.

In this paper, we consider the system in the limit when the length of the time slot goes to zero, similarly as in [6], [8], [14], and analyze the stability of different rate-based (channel-aware) priority scheduling policies.

III. Different policies

Here we present the different schedulers that are analyzed in the paper.

A. Rate-based priority policies

The rate-based schedulers defined below only use information related to the channel state $R_i(t)$ to define the priority $P_i(t)$ of any flow i at time t. A *priority class* consists of flows with the same priority. Ties within a priority class are broken, *e.g.*, randomly or using information of the remaining flow sizes $Y_i(t)$ (if available). Park *et al.* [11] call these schedulers memoryless, since they just utilize the instantaneous values of the rate processes (but not the history).

Definition 1: A scheduling policy π is said to be a *rate-based priority policy* if, for each user class k, there is a strictly increasing real-valued function $h_k(r)$ such that $P_i(t) = h_k(R_i(t))$, for all flows i in class k and all t. Time slot t (with $\mathcal{N}(t) \neq \emptyset$) is allocated to such a flow $i^* \in \mathcal{N}(t)$ for which

$$P_{i^*}(t) = \max_{i \in \mathcal{N}(t)} P_i(t).$$

Let $\mathcal{P}_k$ denote the set of possible priorities of user class k,

$$\mathcal{P}_k = \{h_k(r) : r \in \mathcal{R}_k\}.$$

Denote the highest priority related to user class k by

$$p_k^* = h_k(r_k^*) = \max \mathcal{P}_k,$$

and the second highest priority by

$$p_k^{**} = h_k(r_k^{**}) = \max(\mathcal{P}_k \setminus \{p_k^*\}).$$

Let $\mathcal{P} = \mathcal{P}_1 \cup \ldots \cup \mathcal{P}_K$ denote the *finite* set of all possible priorities. In addition, let $p^* = \max \mathcal{P}$.

Known examples of rate-based priority policies are given below. Before that, we note that if the ties within any *priority class* are broken at random, the priorities $P_i(t)$ within any *user class* k are independently and identically distributed for all $i \in \mathcal{N}_k(t)$ so that each flow of that class gets an equal time share of service.

Borst [6] considered *weight-based priority policies*, for which the priority functions $h_k(r) = w_k r$ are *linear* with weights $w_k > 0$. Time slot t is thus allocated to a flow i^* for which

$$w_{k(i^*)} R_{i^*}(t) = \max_{i \in \mathcal{N}(t)} w_{k(i)} R_i(t),$$

where $k(i)$ refers to the user class of flow i. The weight-based policies that break ties within any *user class* at random are called S^α-strategies in [6]. Borst assumed that sets $\mathcal{P}_k$ are pairwise disjoint, *i.e.*, all priorities are class-specific, which precludes all ties across the user classes. However, we do not make such a restriction in this paper but allow the same priority to appear in any number of sets $\mathcal{P}_k$.

With suitably chosen weights, the highest priority may be given, *e.g.*, for the flow with the highest instantaneous rate, or the flow with the highest instantaneous relative rate (with respect to the mean rate of that class), or the flow with the highest instantaneous proportional rate (with respect to the maximum rate of that class). Next we consider some important special cases for selecting the weights.

The *absolute rate priority policies* are greedy policies maximizing the instantaneous transmission rate. They are weight-based policies with weights $w_k = 1$ for all k so that time slot t is allocated to a flow i^* for which

$$R_{i^*}(t) = \max_{i \in \mathcal{N}(t)} R_i(t). \tag{1}$$

The absolute rate priority policy that breaks ties within any priority class at random is called the MR (Maximum Rate) policy.

The *relative rate priority policies*, instead, are based on the relative user rates. They are weight-based policies with weights $w_k = 1/\bar{r}_k$ so that time slot t is allocated to a flow i^* for which

$$\frac{R_{i^*}(t)}{\bar{r}_{k(i^*)}} = \max_{i \in \mathcal{N}(t)} \frac{R_i(t)}{\bar{r}_{k(i)}}. \tag{2}$$

The idea is to select a user which has good channel quality relative to its expected quality. The RB (Relatively Best) policy introduced in [5] breaks ties within any priority class at random.

Another class of weight-based policies, which we call the *proportional rate priority policies* apply weights $w_k = 1/r_k^*$ so that time slot t is allocated to a flow i^* for which

$$\frac{R_{i^*}(t)}{r^*_{k(i^*)}} = \max_{i \in \mathcal{N}(t)} \frac{R_i(t)}{r^*_{k(i)}}. \tag{3}$$

The idea here is to select a user whose channel quality is good relative to its own best quality. Note that the priority of the maximum rate of user class k is the same for all classes, *i.e.*, $p_k^* = h_k(r_k^*) = 1 = p^*$ for all k. Thus, all the flows with their own maximum rate are in the same highest priority class for sure. The proportional rate priority policy that breaks ties within any priority class at random is called the PB (Proportionally Best) policy.

An example of a rate-based priority policy for which the priority functions $h_k(r)$ are *non-linear* in general is provided by Park *et al.* [11], who introduce the *cumulative distribution function based priority policies* for which $h_k(r) = F_k(r)^{1/w_k}$ with weights $w_k > 0$. Thus, time slot t is allocated to a flow i^* for which

$$F_{k(i^*)}(R_{i^*}(t))^{1/w_{k(i^*)}} = \max_{i \in \mathcal{N}(t)} F_{k(i)}(R_i(t))^{1/w_{k(i)}}. \tag{4}$$

They have the property that the priority of the maximum rate is the same for all user classes, $p_k^* = h_k(r_k^*) = 1 = p^*$ for all k, just like for the proportional rate priority policies. The cumulative distribution function based priority policy that has constant weights $w_k = 1$ for all k and breaks ties within any priority class at random is called the CS (CDF-based Scheduler) policy. It is easy to see that, for the CS policy, the priorities $P_i(t)$ are independently and identically distributed for all $i \in \mathcal{N}(t)$ so that each flow gets an equal time share (in the limit where the time slot length goes to 0). We also note that the CS policy is closely related to the score-based scheduler defined in [15]. In the score-based approach, the user to be scheduled depends on the rank of the instantaneous rate among the rate values of the recent history. The CS policy corresponds to the score-based scheduler with an infinitely long memory.

B. Utility-based policies

At the flow level a natural abstraction of the wireless channel is offered by the throughput experienced by the flows. The scheduler can use, *e.g.*, exponential smoothing to estimate the throughput of different flows. Let $T_i(t)$ denote the throughput of flow i up to time t, and let $U(\theta)$ be a strictly concave utility function of throughput θ. It has been shown in [4] that, for a *fixed set of flows*, the total sum of the utilities, $\sum_i U(T_i(t))$ is maximized (in a certain asymptotic sense with the time constant of exponential smoothing approaching infinity) by a simple gradient-based rule. Based on this, we formulate the following definition for the utility-based scheduling policies (in our wireless context).

Definition 2: A scheduling policy π is said to be a *utility-based policy* if there is a strictly concave and differentiable (utility) function $U(\theta)$ and the decisions are made according to the following gradient-based rule. Time slot t (with $\mathcal{N}(t) \neq \emptyset$) is allocated to such a flow $i^* \in \mathcal{N}(t)$ for which

$$i^* = \arg\max_{i \in \mathcal{N}(t)} R_i(t)\, U'(T_i(t)).$$

Note that the utility-based policies essentially do not have any ties to be broken since the throughput $T_i(t)$ is a continuous quantity (unlike the service rate $R_i(t)$ in our model).

A reasonable general form for the utility function is given by the notion of α-*fairness*, see [10], for which

$$U(\theta;\alpha) = \begin{cases} \log\theta, & \text{if } \alpha = 1, \\ (1-\alpha)^{-1}\theta^{1-\alpha}, & \text{otherwise,} \end{cases}$$

where θ denotes the throughput and $\alpha > 0$ is a fairness parameter.

In particular, if $\alpha = 1$, the utility function corresponds to the well-known *proportionally fair* allocation that asymptotically maximizes $\sum_i \log T_i(t)$ for a fixed set of flows [16]. In this case, the scheduler selects flow i^* for which

$$i^* = \arg\max_{i \in \mathcal{N}(t)} \frac{R_i(t)}{T_i(t)}. \tag{5}$$

The above scheduler is referred to as the PF policy.

Another useful choice is $\alpha = 2$, in which case the utility optimization is equivalent with minimizing asymptotically $\sum_i 1/T_i(t)$ for a fixed set of flows, which is known as minimizing the sum of *potential delays* [12]. Note that the physical unit of the above is equal to s/bit, *i.e.*, a unit somehow related to the flow-level delay. Thus, we define the PD (Potential Delay minimization) policy as a scheduler that selects flow i^* for which

$$i^* = \arg\max_{i \in \mathcal{N}(t)} \frac{R_i(t)}{T_i(t)^2}. \tag{6}$$

IV. Stability results

Here we present the stability results for the model. The results are established using intuitive arguments to give the reader the insight behind the results.

We refer to as channel-aware schedulers all schedulers that utilize instantaneous rate information, as in all the schedulers of Section III. If the base station does not know the instantaneous rates and performs, e.g., simple round robin scheduling (RR), the scheduler is referred to as non-channel aware. Note that using knowledge of remaining flow sizes could still be exploited, cf. use of Shortest Remaining Processing Time (SRPT) as studied in [13].

A. Stability under necessary condition

If there is only one traffic class, i.e., $K = 1$, the situation corresponds to the symmetric setting in [6], and the result states that $\rho^* < 1$ is a sufficient stability condition for linear weight-based strategies. It is easy to see that the same applies in fact to any rate-based priority policy.

Next consider the case when $K \geq 2$. From the classic queueing theory, we know that for any scheduling policy that is *not* channel-aware, such as RR or SRPT, the necessary stability condition is as follows:

$$\rho_1^* \frac{r_1^*}{\bar{r}_1} + \ldots + \rho_K^* \frac{r_K^*}{\bar{r}_K} \leq 1. \tag{7}$$

Borst and Jonckheere [8] showed that for channel-aware policies the necessary stability condition reads as

$$\rho^* = \rho_1^* + \ldots + \rho_K^* \leq 1, \tag{8}$$

which is clearly less stringent than (7), since $r_k^*/\bar{r}_k > 1$ for all k. In addition, Borst and Jonckheere showed that all utility based α-fair policies are stable under condition

$$\rho^* < 1. \tag{9}$$

Below we argue when this condition is sufficient for the stability of rate-based priority policies. As a consequence we find that (9) is a sufficient stability condition, *e.g.*, for the PB and CS policies.

Result 1: Consider a rate-based priority policy π. If $p_k^* > p_l^{**}$ for all $k \neq l$, then π is stable under condition (9).

Proof: We consider the claim in the case $K = 2$. The generalization to the case $K > 2$ is straightforward (but tedious to write out). The argumentation is based on showing that $\rho^* \geq 1$ whenever the system is unstable.

1° Assume first that $p_1^* = p_2^*$. Thus, $p_1^* = p_2^* = p^*$ implying that $R_i(t) = r_{k(i)}^*$ for any flow i with $P_i(t) = p^*$, where $k(i)$ refers to the user class of flow i. Consider now what happens if the system is unstable. Consequently, the number of users grows without limits (as a function of time t), implying that the probability that there is at least one user with the highest priority p^* approaches 1 (as a function of time t). This implies that, in the long run, class-1 traffic is served with rate r_1^* and class-2 traffic with rate r_2^*. Since the system is unstable (and the scheduling policy work-conserving), the proportion of idle slots goes to zero in the long run, which is now equivalent with the requirement that

$$\rho^* = \frac{\lambda_1 \bar{x}}{r_1^*} + \frac{\lambda_2 \bar{x}}{r_2^*} \geq 1.$$

2° Assume now that $p_1^* > p_2^* > p_1^{**}$. Consider what happens if user class 1 is unstable. Consequently, the number of class-1 users grows without limits (as a function of time t), implying that the probability that there is at least one class-1 user with the highest priority $p^* = p_1^*$ approaches 1 (as a function of time t). Since $p_1^* > p_2^*$, this implies that, in the long run, class-1 traffic is served with rate r_1^* while class-2 traffic remains without service. Since the system is unstable (and the scheduling policy work-conserving), the proportion of idle slots goes to zero in the long run, which is now equivalent with the requirement that $\lambda_1 \bar{x}/r_1^* \geq 1$. Consequently,

$$\rho^* > \rho_1^* = \frac{\lambda_1 \bar{x}}{r_1^*} \geq 1.$$

Consider now what happens if user class 1 is stable but class 2 is unstable. Consequently, the number of class-2 users grows without limits (as a function of time t), implying that the probability that there is at least one class-2 user with the highest class-wise priority p_2^* approaches one (as a function of time t). Since $p_2^* > p_1^{**}$, this implies that, in the long run, class-1 traffic is served with rate r_1^* and class-2 traffic with rate r_2^*. Since the system is unstable, the proportion of idle slots goes to zero in the long run, which is now equivalent with the requirement that

$$\rho^* = \frac{\lambda_1 \bar{x}}{r_1^*} + \frac{\lambda_2 \bar{x}}{r_2^*} \geq 1.$$

3° By symmetry, the claim is also valid in the remaining case $p_2^* > p_1^* > p_2^{**}$. ■

In particular, if $p_k^* = p^*$ for all k, then policy π is stable under condition (9), which gives two corollaries below. Note further that the argumentation of Result 1 does not make any assumptions how the possible ties are broken within priority classes. Thus, the stability condition (9) is independent of the tie-breaking rule (as long as it is work-conserving).

Corollary 1: Any proportional rate priority policy (including PB) is stable under condition (9).

Corollary 2: Any cumulative distribution function based priority policy (including CS) is stable under condition (9).

Corollary 3: If $r_k^* = r_J$ for all user classes k, then any absolute rate priority policy (including MR) is stable under condition (9).

Corollary 4: If $r_k^*/\bar{r}_k > r_l^{**}/\bar{r}_l$ for all $k \neq l$, then any relative rate priority policy (including RB) is stable under condition (9).

If we make an additional assumption that ties within a priority class are broken at random, the inequality in Result 1 does not have to be strict, i.e., we get a slightly looser sufficient condition. We state the result here but the detailed argumentation is omitted due to lack of space.

Result 2: Consider a rate-based priority policy π that breaks ties within any priority class at random. If $p_k^* \geq p_l^{**}$ for all $k \neq l$, then π is stable under condition (9).

The argumentation is very similar to the one in Result 1. The intuitive reasoning for the result is that at the stability limit some class k becomes unstable and the number of customers in that class goes to infinity and with high probability there is at least one user with the highest index p_k^*. Then, even though according to Result 2 class-l users may compete with class-k users, the fact that class k is unstable implies that the number of class-k customers goes to infinity and hence the scheduler when making a random selection selects with high probability a user from class-k (remember that class l stays stable and hence the number of class-l users remains finite). This guarantees that at the stability limit class k is served at its own highest achievable rate r_k^*.

B. Further stability conditions for two classes

Here we make a further restriction to derive a sufficient stability condition for the policies that do not satisfy the requirement presented in Result 2. From this on, assume that $K = 2$, ties are broken randomly, and consider the two remaining cases, $p_1^* < p_2^{**}$ and $p_2^* < p_1^{**}$. Since the two cases are symmetric, it is, in fact, sufficient to consider just one of them. Thus, we may assume that $p_2^* < p_1^{**}$.

Let $\tilde{N}_1(t)$ denote the number of flows in a "restricted" system consisting of modified class-1 type flows that arrive with original rate λ_1 and original flow size distribution with mean $\bar{x}$, but the rate processes $\tilde{R}_i(t)$ are restricted from the original processes $R_i(t)$ as follows:

$$\tilde{R}_i(t) = R_i(t) 1_{\{h_1(R_i(t)) > p_2^*\}}.$$

Note that the modified rate processes are still IID stationary and ergodic processes but the stationary distribution is different:

$$\tilde{F}_1(r) = P\{\tilde{R}_1 \leq r\} = \begin{cases} F_1(\tilde{r}_1), & r \leq \tilde{r}_1, \\ F_1(r), & \text{otherwise.} \end{cases}$$

where $\tilde{r}_1 = \max\{r \in \mathcal{R}_1 : h_1(r) \leq p_2^*\}$. In addition, let $\tilde{M}_1(t) = \max_i \tilde{R}_i(t)$. Thus,

$$P\{\tilde{M}_1(t) \leq r \mid \tilde{N}_1(t) = n\} = \tilde{F}_1(r)^n.$$

Note that the conditional probability above is independent of t. In particular, we get

$$P\{h_1(\tilde{M}_1(t)) \leq p_2^* \mid \tilde{N}_1(t) = n\} = F_1(\tilde{r}_1)^n.$$

We assume that the modified system is operated with the same scheduling policy as the original one. Since there are no class-2 flows in this modified system and ties within any priority class are broken at random, the resulting system is a generalized M/G/1-PS queue [17] with state dependent service rates

$$\tilde{G}_1(n) = \frac{1}{\bar{x}} E[\tilde{M}_1(t) \mid \tilde{N}_1(t) = n].$$

Since $\tilde{G}_1(n) \to r_1^*/\bar{x}$, a sufficient condition for the stability of this modified system is $\rho_1^* < 1$. In such a case, the stationary distribution (which is insensitive to the flow size distribution as long as the mean flow size $\bar{x}$ remains the same) is as follows:

$$P\{\tilde{N}_1 = n\} = \frac{\lambda_1^n}{\prod_{l=1}^n \tilde{G}_1(n)} \left(\sum_{m=0}^{\infty} \frac{\lambda_1^m}{\prod_{l=1}^m \tilde{G}_1(m)} \right)^{-1}.$$

Applying now the results presented in [6, Sect. V] to our case $p_2^* < p_1^{**}$, it can be deduced that a sufficient stability condition for S^α-strategies (which are weight-based priority policies) is as follows:

$$P\{h_1(\tilde{M}_1) > p_2^*\} + \rho_2^* < 1, \tag{10}$$

where $P\{h_1(\tilde{M}_1) > p_2^*\} = \sum_{n=0}^{\infty} P\{\tilde{N}_1 = n\}(1 - F_1(\tilde{r}_1)^n)$.

The result can be generalized to any rate-based priority policy that breaks ties within any priority class at random by using similar intuitive arguments as for Results 1 and 2.

Result 3: Assume that $K = 2$, and consider a rate-based priority policy π that breaks ties within any priority class at random.

(i) If $p_2^* < p_1^{**}$, then π is stable under condition

$$P\{h_1(\tilde{M}_1) > p_2^*\} + \rho_2^* < 1. \quad (11)$$

(ii) If $p_1^* < p_2^{**}$, then π is stable under condition

$$\rho_1^* + P\{h_2(\tilde{M}_2) > p_1^*\} < 1. \quad (12)$$

C. Impact of a continuous rate distribution

Now consider the case where the rate processes $R_i(t)$ take values in a continuous but bounded interval (instead of a finite set). From Corollaries 1 and 2, it can be deduced that the proportional rate and CDF-based priority policies are still stable under condition (9). If it is further assumed that the support of the stationary distribution is the same for all user classes k, also absolute rate priority policies are stable under condition (9), as deduced from Corollary 3. However, for the relative rate priority policies a sufficient condition for the stability is more stringent than condition (9).

V. Numerical results

Here we illustrate the properties of the different schedulers in an asymmetric scenario with two user classes, $K = 2$, where we have $J = 11$ possible rates as in the 1xEV-DO system with the rates varying between 38 kbit/s to 2.4 Mbit/s. The following rate-based priority policies are studied: MR, RB, PB, and CS, defined by (1), (2), (3), and (4), respectively. From the utility-based schedulers, we consider the PF and PD policies, defined by (5) and (6), respectively.

For the two user classes, the sets of achievable rates are given by $\mathcal{R}_1(t) = \{r_1, \ldots, r_{j_1}\}$ and $\mathcal{R}_2(t) = \{r_1, \ldots, r_J\}$, *i.e.*, class-2 flows can achieve all $J = 11$ possible rates, while the maximum rate of class-1 flows is limited to $r_{j_1} < r_J$. We assume that the stationary rate distributions of class 1 and 2 flows are truncated geometric distribution with parameters q_1 and q_2, allowing a straightforward parametrization of the rates. Thus, the rate distributions for the two classes are given by

$$P\{R_1 = r_j\} = \frac{q_1^j}{\sum_{n=1}^{j_1} q_1^n}, \quad P\{R_2 = r_j\} = \frac{q_2^j}{\sum_{n=1}^{J} q_2^n}.$$

In the following, we use $q_1 = 1$, $q_2 = 0.5$ and $j_1 = 7$. In this setting, both the MR policy and the RB policy suffer from instability, compared with the necessary condition (9).

A. Simulation implementation

The simulation has been implemented in discrete time and the instantaneous rate of each flow in the system is drawn independently from the given rate distributions for each class. Flows arrive according to a Poisson process with $\lambda_1 = \lambda_2 = 1/2$, and the flow sizes are exponentially distributed. Thus, in the results $E[N] = E[D]$, where D denotes the flow delay. To see the performance, we simulate the system as a function of the load $\rho^* = \rho_1^* + \rho_2^* = (\lambda_1/r_{j_1} + \lambda_2/r_J)E[X]$. Note that to vary the load, since λ_i's are fixed, we vary the mean size of the flows $E[X]$.

The idea is to simulate the system so that the time-scale separation is approximately realized, and hence we used a time slot duration of 0.01 time units, ensuring that a flow completion typically requires many time slots. To verify that the estimates correspond to the stationary behavior of the system even at loads close to the stability limits, the mean delay is estimated directly from the realized delays of the flows that leave the system and also from the time integral of the number of users in the system. We have verified that these two statistics are very close to each other for all loads. In each simulation run, there were at least $3 \cdot 10^5$ arrivals (up to 10^6 for the highest loads) with the initial transient handled by discarding the statistics related to first 10^5 arrivals, and the final results are an average of 10 runs.

In the implementation of the utility-based policies (*i.e.*, PF and PD) we have not used exponential smoothing of the rates to estimate the throughput. Instead, in the simulation at time t we utilize knowledge of the time flow i has been in the system $D_i(t)$ and the amount of bits served from flow i, denoted by $A_i(t)$. Then the throughput of flow i at time t, $T_i(t)$, is directly given by $T_i(t) = A_i(t)/D_i(t)$.

B. Performance of randomized policies

First we consider the overall mean flow delay performance of the basic policies that use randomization to break the ties. The results are shown in Figure 1. In the figure, the left panel gives, as a function of the load ρ^*, the mean flow delay $E[D]$ for the MR, RB and PF policies. The vertical lines (approximately) at $\rho^* = 0.84$, $\rho^* = 0.65$, and $\rho^* = 0.12$ correspond to the stability limits of the MR and RB policies from (12), and that of the non-channel-aware M/G/1 system, respectively. The MR and RB policies both achieve good performance up to a relatively high load as compared with the M/G/1 limit. However, close to the stability limit the delays start rising rapidly. In this case, the MR policy is actually better than PF until the load is close to the MR stability limit. The center panel gives the overall mean delay of the stable policies (PF, PD, CS, and PB) as a function of the load. Clearly, it can be observed that, while the policies are stable up to a very high load, the delay (and the number of customers) is very large. In the center panel, the relative performance of the policies can not be easily seen, and thus in the right panel we have plotted the ratio of the mean delay of PB, CS, and PD to the mean delay of the PF scheduler. It can be observed that, while the PF scheduler is performing best for lower loads, at high loads the PD scheduler is the best (recall that the PD policy minimizes the potential delay, *i.e.*, the mean delay per bit).

Next we study the fairness of the policies. The results are shown in Figure 2, where the ratio of the mean class-2 flow delay to the mean class-1 flow delay is given as a function of the load ρ^* for the MR and RB policies (left panel), the PF and PD policies (center panel), and the PB and CS policies (right panel). The results for MR and RB (left panel) demonstrate that in the beginning the mean class-2 flow delay is greater than for class 1, but, as the stability limit is approached, class-1 delay grows because instability in this case is caused by class

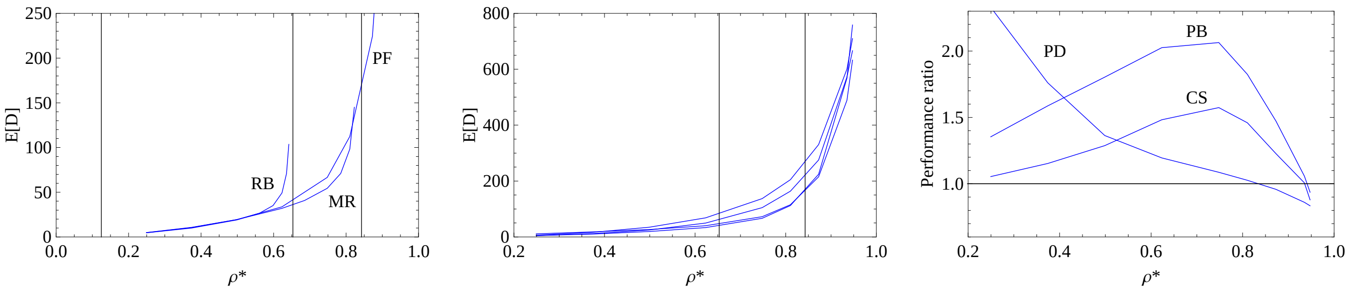

Fig. 1. The mean flow delay $E[D]$ as a function of ρ^* for unstable MR and RB policies (left panel), stable policies (center panel) and the performance ratio of the stable policies relative to PF policy (right panel).

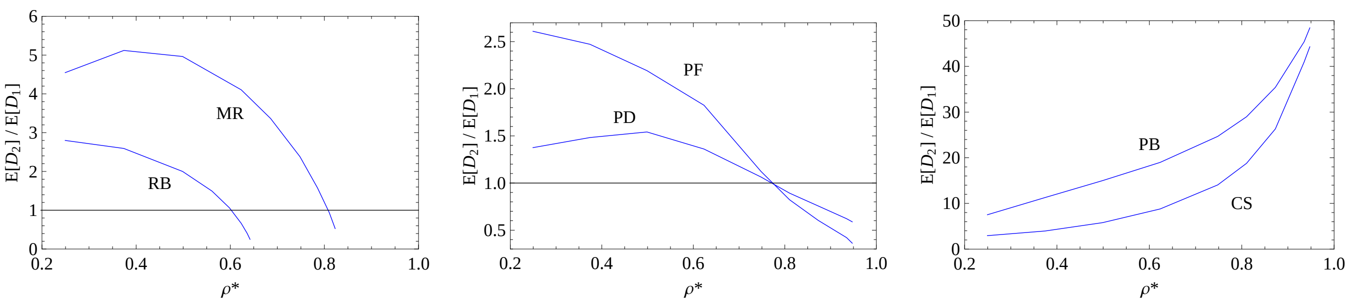

Fig. 2. The ratio of the mean class-2 flow delay to the mean class-2 flow delay $E[D_2]/E[D_1]$ as a function of ρ^* for unstable MR and RB policies (left panel), utility-based policies (center panel), and the PB and CS policies (right panel).

1 becoming unstable. Note that the ratio of the delays stays quite moderate as long as load is not too close to the stability limit. For the utility-based policies, PF and PD (center panel), the delay ratio behaves in a very controlled manner. The delay ratio under the PF policy varies more as the load changes than under the PD policy. Also, both policies are equally fair (ratio equals one) at the same value of the load. Finally, by looking at the results of the PB and CS policies (right panel), it can be observed that, although the policies are stable in the same sense as the PD and PF policies, they are very unfair.

To minimize the flow delays, we also experimented with policies where, instead of random tie breaking, ties are broken using SRPT-like information about the remaining amount of bits left, similarly as in [14]. The results are shown in Figure 3. In the figure, the dashed lines correspond to the randomized basic policies MR and RB, and the corresponding SRPT variant is depicted with a solid line. Also, the straight vertical lines give the stability limits of the M/G/1 queue, RB policy and MR policy, cf. Figure 1. Note that the limits do not hold for the SRPT variants. The results indicate that, even though SRPT is known to be optimal in the M/G/1 queue, in this context it does not necessarily give any benefit (cf., MR and SRPT-MR), leading to the conclusion that utilization of size information at the time slot level is not meaningful.

C. *Optimizing the α-fair policy*

Recall that the utility-based policies can realize bandwidth sharing according to the notion of α-fairness. In the class of

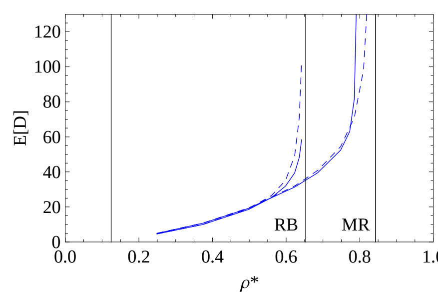

Fig. 3. The performance of SRPT-like tie-breaking heuristics as a function of the load ρ^* for randomized MR and RB policies (dashed line) and their SRPT-like variants (solid lines).

α-fair policies, $\alpha = 0$ corresponds to the MR policy[1], $\alpha = 1$ to the PF policy, $\alpha = 2$ to the PD policy, and $\alpha = \infty$ is the max-min fair policy. Thus, the smaller α is the more aggressively the policy favors users with a high instantaneous rates.

As seen in Figure 1, the PD policy can outperform the PF policy at high loads. The natural question then arises, what is the optimal value of α in the general utility-based policy for a given value of the load? To get some insight to this, simulations were performed with the utility-based scheduler for different values of α and load ρ^*. The results are shown in Figure 4 where for each value of the load we have computed the mean flow delay at a given value of α relative to the

[1]Note that the stability of utility-based policies assumes a strictly concave utility function and $\alpha = 0$ does not satisfy that, see [8].

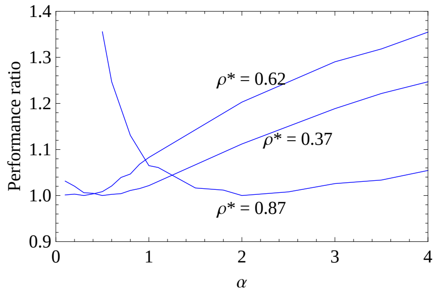

Fig. 4. The performance ratio of α-fair policies as a function of parameter α for three different values of the load ρ^*.

minimum mean delay over the simulated values of α (with the same load). Thus, for each load the performance ratio equals one for the value of α that achieved the smallest delay. From the figure it can be seen that the optimized value of α depends on the load so that for smaller values of load the policy should be more aggressive (minimum is achieved with $\alpha = 0.1$ or $\alpha = 0.5$), while at higher loads there appears to be a minimum at $\alpha = 2$ (the PD policy). The PF policy ($\alpha = 1$) seems to offer an excellent compromise so that it performs well both under low and high loads.

VI. Conclusions

We have analyzed the stability properties of channel-aware priority-based policies. These policies represent a generalization of many well known channel-aware policies, including the weight-based strategies in [6]. The general prerequisites when the policies are stable under the necessary condition have been derived, which holds for arbitrary tie-breaking policies within a class and is independent of the flow-size distribution. Also, when the requirements of the necessary condition are not met, a more stringent sufficient condition has been derived for the case with two user classes.

We also made simulations of certain priority-based policies with randomized tie breaking (MR, RB, PB and CS) and two utility-based policies (PF and PD) to investigate their performance. The results showed that MR and RB perform quite well but become unstable well before the limit corresponding to the necessary condition. While, the PB and CS policies are stable up to the necessary limit, their overall performance is quite poor and unfairness is a major issue. The PF and PD policies are both stable policies up to the necessary limit. For a wide range of loads, the PF policy performs better. Only at very high loads the PD scheduler gives slightly better performance than PF. Our conclusion is that the PF policy is a robust policy which provides very good fairness properties and overall performance across all values of the load. Experiments with SRPT-like heuristics for tie breaking revealed that applying size information at the time slot level does not necessarily improve performance.

As an on-going work, we are strengthening the intuitive arguments used to prove the results in this paper by a more formal mathematical treatment. A further fundamental open problem is how to minimize the flow level delay by using size information. We believe that the proper approach for using flow-level information on sizes is to use it to parameterize the packet-level scheduler (cf., results in Section V-C) so as to minimize a given performance criterion in a similar manner as α-fair policies optimize a certain utility function. Obviously the objective function should be then directly related to the flow-level delay.

Acknowledgement

This research has been partially supported by the AWA (Advances in Wireless Access) project, funded by Ericsson, Nokia-Siemens Networks and TEKES. Also, fruitful discussions with Prof. Sem Borst are gratefully acknowledged.

References

[1] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1277–1294, jun 2002.
[2] A. Stolyar, "Maxweight scheduling in a generalized switch: state space collapse and equivalent workload minimization in heavy traffic," *Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, 2004.
[3] P. Liu, R. Berry, and M. Honig, "A fluid analysis of a utility-based wireless scheduling policy," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2872–2889, July 2006.
[4] A. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation," *Operations Research*, vol. 53, no. 1, pp. 12–25, 2005.
[5] F. Berggren and R. Jäntti, "Asymptotically fair transmission scheduling over fading channels," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 326–336, Jan. 2004.
[6] 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.
[7] M. Andrews, K. Kumaran, K. R. A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, no. 2, pp. 191–217, 2004.
[8] S. Borst and M. Jonckheere, "Flow-level stability of channel-aware scheduling algorithms," in *Proc. of WiOpt*, April 2006, pp. 1–6.
[9] S. Borst, "Flow-level performance and user mobility in wireless data networks," *Philosophical Transactions of the Royal Society*, vol. 366, pp. 2047–2058, 2008.
[10] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.
[11] D. Park, H. Seo, H. Kwon, and B. Lee, "Wireless packet scheduling based on the cumulative distribution function of user transmission rates," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1919–1929, nov 2005.
[12] L. Massoulié and J. Roberts, "Bandwidth sharing: objectives and algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 320–328, 2002.
[13] 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.
[14] P. Lassila and S. Aalto, "Combining opportunistic and size-based scheduling in wireless systems," in *Proc. of ACM MSWiM*, 2008, pp. 323–332.
[15] T. Bonald, "A score-based opportunistic scheduler for fading radio channels," in *Proc. of European Wireless*, Feb. 2004, pp. 283–292.
[16] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
[17] J. Cohen, "The multitype phase service network with generalized processor sharing," *Acta Informatica*, vol. 12, pp. 245–284, 1979.