Flow Size-Aware Proportional Fair Scheduler

Kinda Khawam, Daniel Kofman
{khawam, kofman}@enst.fr
GET/ENST - Telecom Paris
46, Rue Barrault
75013 Paris, France

***Abstract*— In new generation cellular networks, opportunistic schedulers take advantage from the delay-tolerance of data applications to ensure that transmission occurs when radio channel conditions are most favourable. "Proportional Fair" (PF) is a well-known opportunistic scheduler that provides a good compromise between fairness and efficiency when transmitting long flows. Unfortunately, the PF algorithm is not efficient for short transfers, which represent the majority of data flows. Moreover, the lack of coordination between opportunistic scheduling and the congestion control mechanism of TCP induce very poor performance especially for short-lived flows. In this paper, we propose three enhanced scheduling approaches that significantly reduce the transfer time of short flows without a significant degradation of the QoS provided to long flows.**

I. Introduction

Proportional fair scheduling has been largely studied in the context of HDR(High Data Rate)/HDSPA (High Data Speed Packet Access) based mobile networks [1], [2]. Despite all the benefits resulting from opportunistic scheduling, it has also been shown that the lack of coordination between the latter and the congestion control mechanism of TCP induce very poor performances [3]. In the cited paper, the authors propose an enhancement to PF to deal with this issue when transmitting long TCP flows. Nevertheless, the proposed approach does not help in enhancing the QoS provided to short flows which are the most adversely impacted by the mentioned lack of coordination while they require the shortest response time. This is critical since, as it has been shown by various Internet traffic analyses [9], short flows represent the vast majority of Internet flows. What is more, the PF algorithm is biased against short-lived flows because it can not converge when the flow lasts a few time slots resulting in suboptimal performances especially at high loads.

In order to improve short flows performance, we propose in this paper three flow size-aware scheduling mechanisms where file size information is integrated in the PF algorithm to reduce the Sojourn Time experienced by short flows. The proposed approaches notably reduce the transfer time of short-lived flows, without a significant degradation of the QoS provided to long-lived flows.

The rest of the paper is organized as follows. In section II, we present the radio propagation model and our traffic hypothesis. In section III, we show how the PF is biased against short-lived flows. In sections IV to VI, we present our three flow size-aware scheduling mechanisms along with analytical study and simulations to assess performances. We give conclusions in section VII.

II. Radio Propagation and Traffic Models

In this section, we present the model of the radio resource and the way it is shared among users. We then describe our adopted hypothesis relative to the offered traffic.

A. The Propagation Model

Let P be the transmission power of the Base Station (BS), γ the free space path loss and x_k the fading (of unit mean) for user k. The power received by user k situated at distance r from the BS is then given by:

$$P_k(r,t) = P \cdot \gamma(r) \cdot x_k(t) \tag{1}$$

The adopted model for the free space path loss is:

$$\gamma = 1 \text{ if } r \leq \epsilon \text{ and } \gamma = (\tfrac{\epsilon}{r})^{\beta} \text{ otherwise}$$

where β is the path loss exponent (taking values between 2 and 5) and ϵ is the maximum distance at which the full power P is received.

B. The feasible rate

The feasible rate R of a user depends on the radio channel and varies with time due to user mobility and fading effects. The mobility will not be included in the model, nor will the slow fading be.
We denote by C_0 the maximum peak rate offered by the used coder and by r_0 the maximum distance at which this peak rate is achieved, i.e., $r \leq r_0 \Leftrightarrow R = C_0$.
For user k, the signal-to-noise ratio and energy-per-bit to noise density ratio [4] are respectively equal to:

$$SNR_k = \frac{P_k}{(\eta + I_k)}, \frac{E_b}{N_0} = \frac{W}{R} \cdot SNR_k \tag{2}$$

where W is the cell bandwidth, P_k the power received by user k, η the background noise and I_k the interference due to other BSs.
For a given target error probability, $\frac{E_b}{N_0}$ must be greater than a given threshold σ (taken as a constant as it is done in the majority of references), the feasible data rate of user k is then given by:

$$R_k = min[C_0, \frac{W}{\sigma} \cdot SNR_k] \tag{3}$$

We consider hexagonal networks, thus the interference suffered by a user in a given cell is almost utterly generated by

0-7803-9455-0/06/$20.00 (c) 2006 IEEE

the 6 neighbouring BSs. An approximation of this interference is given by the following [5]:

$$\begin{aligned} I(r) =& P \cdot \big[\gamma(2\Re - r) + 2 \cdot \gamma\left(\sqrt{(\Re - r)^2 + 3\Re^2}\right) + \\ & \gamma(2\Re + r) + 2 \cdot \gamma\left(\sqrt{(\Re + r)^2 + 3\Re^2}\right)\big] \end{aligned} \tag{4}$$

where $\Re$ is the cell radius that we suppose equal to $2 \cdot r_0$ (larger values of $\Re$ induce very small rates at the border of the cell [5]).

Combining equations (1), (3) and (4) , we have the subsequent result for negligible background noise:

$$\begin{aligned} R_k(r,t) =& min\big[C_0, \frac{W}{\sigma} \cdot (P \cdot \gamma(r) \cdot x_k(t) \div I(r))\big] \\ =& min\big[C_0, \frac{W}{\sigma} \cdot h\left(\frac{\Re}{r}\right) \cdot x_k(t)\big] \end{aligned} \tag{5}$$

where $h^{-1}(\frac{\Re}{r}) = (2\frac{\Re}{r} - 1)^{-\beta} + 2 \cdot \big((\frac{\Re}{r} - 1)^2 + 3(\frac{\Re}{r})^2\big)^{-\frac{\beta}{2}} + (2\frac{\Re}{r} + 1)^{-\beta} + 2 \cdot \big((\frac{\Re}{r} + 1)^2 + 3(\frac{\Re}{r})^2\big)^{-\frac{\beta}{2}}$

We know that the PF system does not behave like a GPS queue if the feasible rate $R_k(r,t)$ does not vary linearly with the i.i.d fast fading fluctuations $x_k(t)$ for all users, which is not the case if there are users located at a distance $r \leq r_0$. For that reason, we only consider users located between $r_0 < r \leq \Re$ with homogeneous fading and the following feasible rate:

$$R_k(r,t) = C_k(r) \cdot x_k(t) \tag{6}$$

with

$$C_k(r) = \frac{W}{\sigma} \cdot h\left(\frac{\Re}{r}\right) \tag{7}$$

being the mean rate of user k. As we consider Rayleigh fading, x_k follows an exponential distribution.

C. Traffic characteristics

We assume traffic demand is uniformly distributed in the cell. Data flows arrive as a Poisson process of intensity $\lambda \cdot ds$ in any area of surface ds. This assumption is fairly plausible since traffic is due to the independent activity of a large population of users, each individually having a very small intensity, which can therefore be modeled by a Poisson process.
Flow size σ follows the Bounded Pareto (BP) distribution which is commonly used in analysis because it can exhibit the high variance property as observed in the internet traffic and also because the maximum flow size can be set to mimic the largest internet flow size. We denote the BP distribution by $BP(p,q,\alpha)$ where p and q are respectively the minimum and maximum flow size and α is the exponent of the power law. The probability density function of the BP distribution is:

$$f(x) = \frac{\alpha \cdot \left(\frac{p}{C}\right)^{\alpha}}{1 - \left(\frac{p}{q}\right)^{\alpha}} \cdot x^{-\alpha-1}, \quad p \leq x \leq q, 0 \leq \alpha \leq 2$$

where C is the mean rate of the flow.
We will consider the following scenario:

- 80% of flows are short flows whose size follows the following distribution $BP(1kbytes, 10kbytes, 1.16)$ of mean $\mathbb{E}[\sigma_S] = 2.4kbytes$.
- While 20% of flows are long flows whose size follows the following distribution $BP(10kbytes, 5000kbytes, 1.16)$ of mean $\mathbb{E}[\sigma_L] = 45.7kbytes$.

Thus the load of short flows is approximately 17.5% and the load of long flows is approximately 82.5%. Hence, we have two classes of flows and we denote by $d\rho_S = 2\pi r dr \cdot \lambda_S \mathbb{E}[\sigma_S]$ and by $d\rho_L = 2\pi r dr \cdot \lambda_L \mathbb{E}[\sigma_L]$ the traffic intensity respectively generated by short flows and long flows whose distance to the BS is between r and $r+dr$, where $\rho_S = 0.8 \cdot \rho$ and $\rho_L = 0.2 \cdot \rho$.

III. Proportional Fair algorithm

In the PF algorithm, time is divided into short intervals and the BS transmits at full power to a single user per time slot. At time slot t, the scheduled user is the one with the highest feasible rate relative to its current average throughput, i.e.,

$$\text{user } k^* = arg_k max \frac{R_k(r,t)}{T_k(r,t)}$$

where $T_k(r,t)$ is the exponentially smoothed throughput:

$$T_k(r,t+1) = (1 - \frac{1}{\tau}) \cdot T_k(r,t) + \frac{1}{\tau} \cdot R_k(r,t) \cdot \mathbb{1}_{user(t)=k} \tag{8}$$

where $\mathbb{1}_{user(t)=k}$ is the indicator function which equals 1 if user k was chosen at time slot t and 0 otherwise. τ is a time constant that captures the time-scales of the PF scheduler. Because the random variables representing the fading are i.i.d, we have that $T_k = C_k(r) \cdot U_k$ where U_k are identically distributed random variables (but not independent).
If $\frac{1}{\tau} \to 0$, then

$$T_k \to C_k(r) \cdot \frac{g(n)}{n} \tag{9}$$

where n is the total number of active users and $g(n) = \mathbb{E}[max(x_1, .., x_n)]$ is the PF scheduling gain. In practice, τ has large values because this offers the opportunity of waiting a long time before scheduling a user when its channel quality is maximal: the scheduler is then expected to better exploit multi-user diversity. As a result, we will adopt formula (9) in our analysis. We refer to [6] for rigorous justifications of the above claims. Therefore, the average rate of user k is:

$$\begin{aligned} & \mathbb{E}[R_k(r,t) \cdot \mathbb{1}\{\frac{R_k(r,t)}{\frac{g(n)}{n} \cdot C_k(r)} = max_{l=1..n} \frac{R_l(r,t)}{\frac{g(n)}{n} \cdot C_l(r)}\}] \\ & = C_k(r) \cdot \mathbb{E}[x_k(t) \cdot \mathbb{1}\{\frac{x_k(t) \cdot n}{g(n)} = max_{l=1..n} \frac{x_l(t) \cdot n}{g(n)}\}] \\ & = C_k(r) \cdot \mathbb{E}[x_k \cdot \mathbb{1}\{x_k = max_{l=1..n} x_l\}] \\ & = \frac{C_k(r)}{n} \cdot \mathbb{E}[max(x_1, ..x_n)] = \frac{C_k(r) \cdot G(n)}{n} \end{aligned}$$

We see through the realized average rate that when the PF algorithm converges, it behaves like a GPS system while taking advantage from the channel variations with $G(n) = \mathbb{E}[max(x_1, ..x_n)]$ being the multi-user diversity gain (which is an increasing function in the number of active users n). In the case of Rayleigh fading, we have $G(n) = \sum_{i=1}^{n} \frac{1}{i}$.

However, the PF is not quite fair and efficient when serving short-lived flows which is a real issue considering that these flows represent the majority of elastic traffic. Indeed, the PF performance is biased against short flows because the algorithm does not have time to converge when the flow lasts a few time slots. To highlight this fact, we give in the following section the analysis of the dynamic model of PF, in particular we give the analytical expressions for the mean sojourn time of short flows and the mean throughput of long flows and underline the discrepancy, at high loads, between the mathematical model and simulation results for short flows.

A. The Analytical Study of PF

When convergence is reached, PF behaves like a GPS system, we can thus compute the mean transfer delay for short flows and mean throughput for long flows according to [7].
The stationary distribution of the number of active flows is: $\pi(x) = \frac{\prod_{i=1}^{x} \frac{\rho}{G(i)}}{\sum_{k=0}^{n} \prod_{i=1}^{k} \frac{\rho}{G(i)}}$ where $\rho = \int_{r_0}^{\Re} \frac{d\rho(r)}{C(r)}$ is the load in the cell with $d\rho(r) = d\rho_S(r) + d\rho_L(r)$ and n is the maximum number of admitted flows. $C(r)$ follows from (7).
Using Little's law, the mean transfer delay experienced by a short flow k is $\mathbb{E}[S_{S,k}] = \frac{\mathbb{E}[\sigma_S] \cdot \mathbb{E}[n]}{C_k(r) \cdot \rho \cdot (1-B)}$, where $B = \pi(x = n)$ is the blocking probability and $\mathbb{E}[n] = \sum_{i=1}^{n} i \cdot \pi(i)$ is the mean number of active flows.
The throughput of a long flow k, defined as the ratio of the mean long flow size $\mathbb{E}[\sigma_L]$ to the mean long flow duration, is $Th_{L,k} = \frac{\rho \cdot (1-B) \cdot C_k(r)}{\mathbb{E}[n]}$.

B. Numerical experiments

We present in this subsection our numerical experiments performed to illustrate the previous results. We consider a system where users initiate file transfer requests as a Poisson process of intensity $\lambda \pi \Re^2$. Flow sizes are independent and follow the composite heavy-tailed distribution presented in section II-C. Users are served according to the PF algorithm and at most $n = 40$ users are admitted in the system to guarantee a minimum rate of $C_{min} = \frac{C(\Re)}{40}$. Guaranteeing a minimum rate is a QoS notion appropriate for non-real time users. New transfers generated when the maximum number of users is already in progress are blocked and lost. We take $\frac{W}{\sigma} = 5.0$. We determine the normalized mean sojourn time for short flows $\mathbb{E}[S_{S,k}] \cdot C_k(r)$ depicted in Figure 1 and the normalized mean throughput $\frac{Th_{L,k}}{C_k(r)}$ for long flows depicted in Figure 2.
We can see in Figure 2 how the analytical formulae provide highly accurate estimates of results obtained by simulation for long flows which is not the case when $\lambda \geq 3.0$ ($\rho_S \geq 0.45$) for short flows as graphed in Figure 1. Indeed, in PF, it is impractical to guarantee the same probability of accessing the channel for all flows over short time scales; yet, over longer time scales, as channel conditions vary, lagging flows can "catch up" which is not possible for short flows. This situation is, of course, further exacerbated at high loads when the cell is crowded because short flows have to wait longer for their turn which can have a severe impact on their Sojourn Time. This problem is more emphasized by the large value taken by τ in

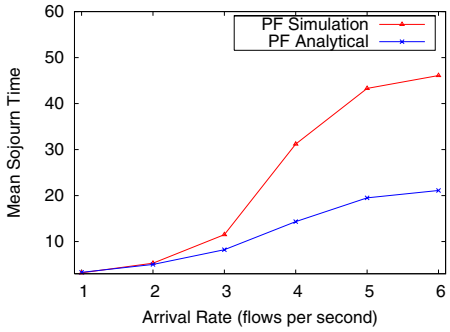

Fig. 1
MEAN SOJOURN TIME OF SHORT FLOWS

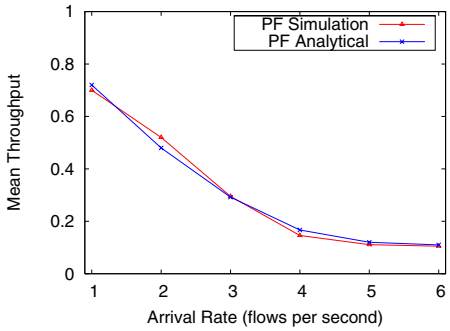

Fig. 2
MEAN THROUGHPUT OF LONG FLOWS

(8). However, reducing the value of τ is not an appropriate solution because it deprives us from better exploiting multi-user diversity. To overcome this inherent drawback of PF, we propose in this paper three modified versions of PF where short flows are given preferential treatment. We suggest in the first algorithm, termed Size-Based Hierarchical PF (SB-HPF), to isolate short flows from long flows, in order to protect the former. Slots are first distributed among the two classes of flows and inside each class, flows are served according to the PF algorithm. In the second algorithm, termed Size Based Adapted PF (SB-APF), we tweak the PF algorithm in a way to augment the probability of short flows to access the channel. The last algorithm, termed PF-LAS, is a unified version of both PF and the LAS (Least Attained Service) algorithm, which is known to favor short flows to the few largest flows.

IV. SB-HPF

Our hierarchical scheduler serves alternately short flows (with a weight w_S) and long flows (with a weight w_L) that are logically separated into two classes and applies independent PF to each class.

A. The Analytical Study of SB-HPF

Because the two classes of flows, served by means of PF, behave like a GPS system, we can compute the average

transfer delay for short flows and the average realized throughput for long flows. The stationary distribution of the number of active flows in class Z is: $\pi_Z(x) = \frac{\prod_{i=1}^{x} \frac{\rho_Z}{G(i)}}{\sum_{k=0}^{n_Z} \prod_{i=1}^{k} \frac{\rho_Z}{G(i)}}$ where $\rho_Z = \int_{r_0}^{\Re} \frac{d\rho_Z(r)}{C(r) \cdot w_Z}$ is the load in class Z and n_Z is the maximum number of admitted flows in that class. From Little's law, the mean transfer delay experienced by a short flow k is $\mathbb{E}[S_{S,k}] = \frac{\mathbb{E}[\sigma_S] \cdot \mathbb{E}[n_S]}{C_k(r) \cdot w_S \cdot \rho_S \cdot (1 - B_S)}$ where $B_S = \pi_S(x = n_S)$ is the blocking probability for short flows and $\mathbb{E}[n_S] = \sum_{i=1}^{n_S} i \cdot \pi_S(i)$ is the mean number of active short flows.

The throughput $Th_{L,k}$ of a long flow k, defined as the ratio of the mean long flow size $\mathbb{E}[\sigma_L]$ to the mean long flow duration, is $Th_{L,k} = \frac{\rho_L \cdot (1 - B_L) \cdot C_k(r) \cdot w_L}{\mathbb{E}[n_L]}$ where $B_L = \pi_L(x = n_L)$ is the blocking probability for long flows and $\mathbb{E}[n_L] = \sum_{i=1}^{n_L} i \cdot \pi_L(i)$ is the mean number of active long flows.

The allocation of slots being dynamic, the proposed scheduler will guarantee a minimum throughput $Th_{L,k}$ and a maximum sojourn time $\mathbb{E}[S_{S,k}]$ for each flow k.

1) Numerical experiments: We present in this section our numerical experiments performed with the same parameters as those presented in section III-B. Users are served according to PF ($n = 40$) and according to SB-HPF that comprises 2 different scenarios:

- SB-HPF(2,1) where short flows receive 2 times more slots than long flows, i.e.,$w_S = 2/3$ and $w_L = 1/3$. To obtain the same minimum rate as in PF ($C_{min} = \frac{C(\Re)}{40}$), at most 26 short flows ($C_{min} = \frac{C(\Re)}{26} \cdot \frac{2}{3}$) and 14 long flows ($C_{min} = \frac{C(\Re)}{14} \cdot \frac{1}{3}$) are admitted simultaneously.
- SB-HPF(1,2) where long flows receive 2 times more slots than short flows, i.e.,$w_L = 2/3$ and $w_S = 1/3$. To obtain the same minimum rate as in PF, at most 26 long flows ($C_{min} = \frac{C(\Re)}{26} \cdot \frac{2}{3}$) and 14 short flows ($C_{min} = \frac{C(\Re)}{14} \cdot \frac{1}{3}$) are admitted simultaneously.

In all experiments, we determine the normalized average sojourn time for short flows $\mathbb{E}[S_{S,k}] \cdot C_k(r)$ and the normalized average throughput for long flows $\frac{Th_{L,k}}{C_k(r)}$. Figures 4 and 6 illustrate the mean sojourn time for short flows as a function of arrival rate and indicate that the analytical formulae provide a highly accurate estimate for SB-HPF, which proves that the PF converges well for short flows when they are separated from long flows. Figures 5 and 7 illustrate the mean throughput for long flows as a function of arrival rate and indicate that the analytical formulae give a lower bound on simulation results at low loads, while at high loads, the two sets of values coincide as predicted. As for results, we can see that short flows profit largely from being isolated from long flows for $\lambda \geq 3.0$ because there are relieved from the bias in the behaviour of PF. For smaller values of λ, the mean number of short flows is strictly smaller than 1 in SB-HPF, which means that there are rarely more than two short flows present simultaneously and thus they do not profit from the gain resulting from multi-user diversity. For the same values of λ, there are more flows in the system served according to the original PF and so short flows profit from the benefits of opportunistic scheduling without its mentioned adverse effects in a crowded cell. The gain obtained for short flows at high load comes at the expense of long flows who will see a degradation in their performances in terms of mean throughput. Nevertheless, the degradation is not severe at low loads because long flows profit from the surplus of

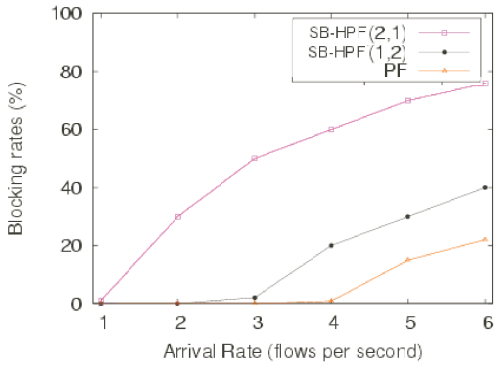

Fig. 3
BLOCKING RATES FOR LONG FLOWS

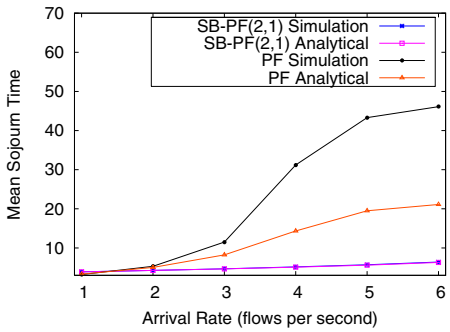

Fig. 4
MEAN SOJOURN TIME OF SHORT FLOWS FOR SB-HPF(2,1)

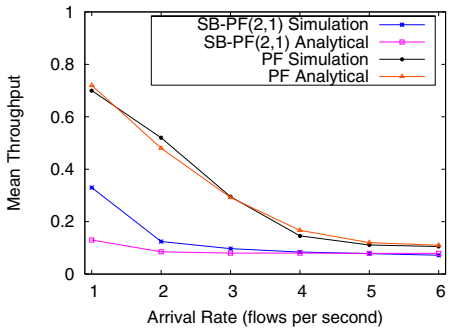

Fig. 5
MEAN THROUGHPUT OF LONG FLOWS FOR SB-HPF(2,1)

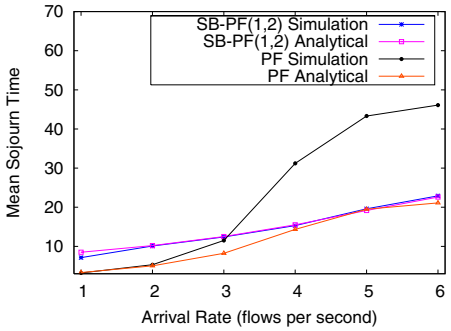

Fig. 6

MEAN SOJOURN TIME OF SHORT FLOWS FOR SB-HPF(1,2)

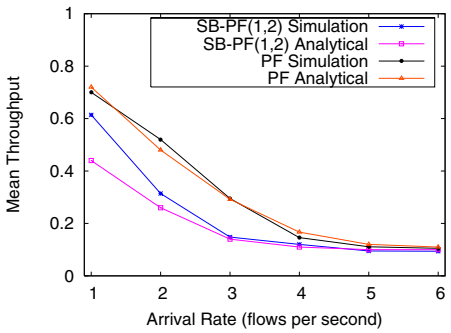

Fig. 7

MEAN THROUGHPUT OF LONG FLOWS FOR SB-HPF(1,2)

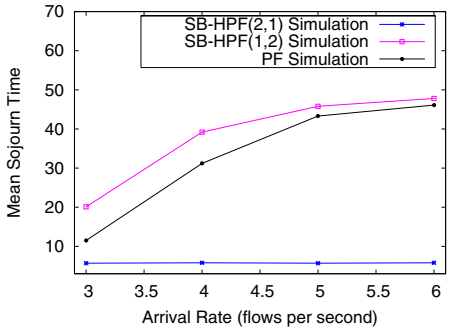

Fig. 8

MEAN SOJOURN TIME OF SHORT FLOWS

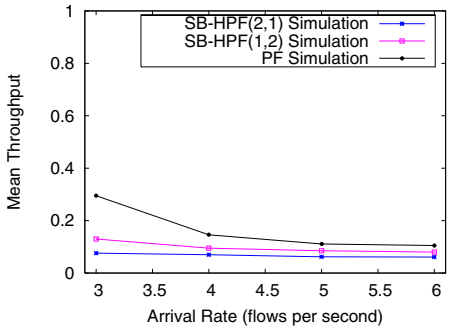

Fig. 9

MEAN THROUGHPUT OF LONG FLOWS

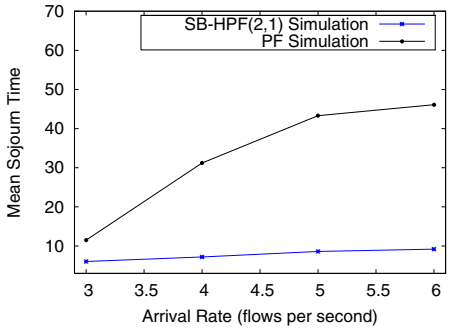

Fig. 10

MEAN SOJOURN TIME OF SHORT FLOWS (WITH MBAC)

slots reserved to short flows. As for high loads, although this deterioration in performances is limited by admission control, it comes at the cost of relatively high blocking rates as depicted in Figure 3. However, in SB-HPF(1,2), long flows perceive only a slight degradation in terms of mean throughput at a cost of acceptable blocking rates at high loads in comparison with the original PF. As for short flows, we can see in Figure 4 that for $\lambda \geq 3.0$, they realize much lower transfer delays in comparison with PF. We conclude that SB-HPF(1,2) is beneficial at high loads, exactly when the PF falls short from treating fairly short flows, while still preserving the performances of long flows.

In what preceded, we supposed that we had an ideal knowledge of the size of each flow, yet, this is not the case in the real world. Since the majority of data flows are short flows, we will run another set of experiments where we will start off optimistically by considering that every new flow is a short flow. If a flow turns out to be a long flow (the number of packets received for this flow goes beyond a certain threshold (10Kbytes)), then the system switches it to the class of long flows. As we can see from Figure 8, short flows suffer from increasing transfer delay in SB-HPF(1,2) as compared with PF. This was expected as the main benefit of the proposed algorithm vanishes when long and short flows are initially mixed. The bias of PF against short flows reappear because they are no longer isolated from long flows. However, in SB-HPF(2,1), short flows are served frequently enough so that the PF algorithm converges even when they are served along with long flows. Their transfer delays are notably lower than what they obtain in plain PF. As for long flows, we can see in Figure 9 that the degradation in term of mean throughput is acceptable in comparison with PF, but more importantly these performances are obtained for blocking rates that are comparable with those of PF. We conclude that when the size of flows is not known a priori, the SB-

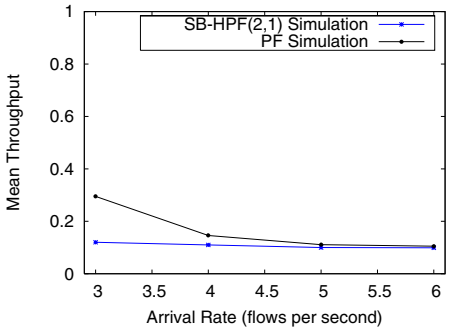

Fig. 11

MEAN THROUGHPUT OF LONG FLOWS (WITH MBAC)

HPF(2,1) scenario lowers remarkably the mean Sojourn Time of short-lived flows without penalizing long-lived flows too much. Nevertheless, we can still remedy this slight degradation in performance for long flows through Measurement-Based Admission Control (MBAC). In MBAC, it is the performance of long flows that will trigger the blocking of new arrivals because of the discrimination performed against these flows. Since the number of ongoing flows is limited, the system will easily compute the throughput realized by each long flow without facing scalability issues, and whenever any of these throughputs goes below a predefined minimal value, Th_{min}, all new arrivals are blocked. New flows are admitted back in the system if all throughputs realized by long flows go beyond another predefined value ($\delta \cdot Th_{min}$ with $\delta > 1$). For each λ, Th_{min} will be set equal to the throughput obtained for long flows by the analytical formula for SB-HPF(2,1). The improvement obtained for long flows comes at the price of a slight increase in blocking rates. We have thus significantly favoured short flows without penalizing long flows.

V. SB-APF

We know that the PF scheduler serves the user with the highest feasible rate relative to its current average throughput (this ratio is commonly called RICQ) and so in order to favour short flows, we propose to increase their RICQ (or to reduce the RICQ of long flows) thus increasing their chance to be selected by the scheduler. In order to do that, two methods are put forward: the first one is termed SB-APF-a and the second SB-APF-b.

A. SB-APF-a

Aiming to increase the RICQ of short flows in comparison with long flows, we chose in this algorithm to divide the RICQ of the long flows by a constant A greater than 1. Thus short flows will have more chance to be picked by the scheduler in comparison with long flows.

1) Analytical Study: Long flows will be indexed by 0 and short flows by 1. The exponential throughput of a long flow i is $T_{0,i} = \frac{A \cdot g(n_0)}{n_0} \cdot C_{0,i}$ and of a short flow i is $T_{1,i} = \frac{g(n_1)}{n_1} \cdot C_{1,i}$. The number of long and short flows are respectively n_0 and n_1. The average rate of a long flow i is:

$$\begin{aligned}
&\mathbb{E}[R_{0,i} \cdot \mathbb{1}\{\frac{R_{0,i}}{T_{0,i}} = max_{k=0,1} max_{l=1,..,n_k} \frac{R_{k,l}}{T_{k,l}}\}] \\
=&C_{0,i} \cdot \mathbb{E}[x_{0,i} \cdot \mathbb{1}\{\frac{x_{0,i} \cdot n_0}{A \cdot g(n_0)} \geq \frac{n_1}{g(n_1)} \cdot Z_1\} \cdot \mathbb{1}\{x_{0,i} = Z_0\}] \\
=&\frac{C_{0,i}}{n_0} \cdot \mathbb{E}[Z_0 \cdot \mathbb{1}\{Z_0 \geq \frac{A \cdot n_1 \cdot g(n_0)}{g(n_1) \cdot n_0} \cdot Z_1\}]
\end{aligned} \tag{10}$$

with $Z_k = max(x_{k,1}, .., x_{k,n_k})$.

We denote by $f_k(z)$ the density function of Z_k. The random variables x_k having an exponential distribution of unit mean, we get $f_k(z) = n_k \cdot e^{-z} \cdot (1 - e^{-z})^{n_k - 1}$. Thus (10) yields $\frac{C_{0,i}}{n_0} \cdot \int_0^\infty f_1(x) \int_{\frac{A \cdot g(n_0) n_1}{g(n_1) n_0} \cdot x}^\infty y \cdot f_0(y) dy dx$.

Another value of interest to our algorithm is the probability of short flows to access the channel because increasing this probability comes down to reducing the mean sojourn time of short flows. The conditional probability that a short flow i has a relatively better channel than a long flow j is:

$$\mathbb{P}(\frac{R_{0,j}}{T_{0,j}} \leq \frac{R_{1,i}}{T_{1,i}} | R_{1,i}) = 1 - e^{-A \frac{g(n_0) \cdot n_1}{g(n_1) \cdot n_0} \cdot x}$$

The resulting asymptotic time fraction assignment $P_{1,i}$ for a short flow i is given by:

$$\begin{aligned}
P_{1,i} &= \int_0^\infty \big[(\mathbb{P}(\frac{R_{0,j}}{T_{0,j}} \leq \frac{R_{1,i}}{T_{1,i}} | R_{1,i}))^{n_0} \\
&\cdot \big(\mathbb{P}(\frac{R_{1,k}}{T_{1,k}} \leq \frac{R_{1,i}}{T_{1,i}} | R_{1,i})\big)^{n_1 - 1} \cdot e^{-x} dx\big] \\
&= \int_0^\infty (1 - e^{-A \frac{g(n_0) \cdot n_1}{g(n_1) \cdot n_0} \cdot x})^{n_0} \cdot (1 - e^{-x})^{n_1 - 1} \cdot e^{-x} dx \\
&= \sum_{k=0}^{n_0} \sum_{l=0}^{n_1 - 1} C_k^{n_0} C_l^{n_1 - 1} (-1)^{k+l} (1 + l + kA \frac{g(n_0) n_1}{g(n_1) n_0})^{-1}
\end{aligned}$$

For $A = 1.0$, $n_0 = n_1$ and $n = n_0 + n_1$, we get $P_{1,i} = \frac{1}{n}$ and thus we have the original PF, but for $A > 1.0$, $P_{1,i} > \frac{1}{n}$ while the asymptotic time fraction assignment for a long flow j, $P_{0,j}$, is smaller than $\frac{1}{n}$. We conclude that SB-APF-a increases the chance of short flows to access resources and hence reduces their transfer times.

2) Numerical Experiments: We present in this section our numerical experiments performed to illustrate the previous results. We set $n_0 = n_1 = 20$ in SB-APF-a and $n = n_0 + n_1 = 40$ in PF. Users are served according to PF and according to our SB-APF-a that is run for different values of A that ranges from 1.1 to 2.5. We determine in the first set of experiments the probability to access the channel for short flows and compare it to $P_{1,i}$. We determine in the second set of experiments the normalized average rate for long flows and compare to (10) divided by $C_{0,i}$.

The results as graphed in Figures 12 and 13 show that the analytical formulae give pretty exact estimates of the values obtained by simulation. However, the formulae in SB-APF-a

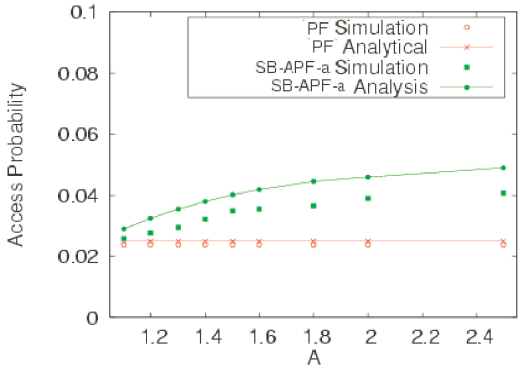

Fig. 12

ACCESS PROBABILITY OF SHORT FLOWS

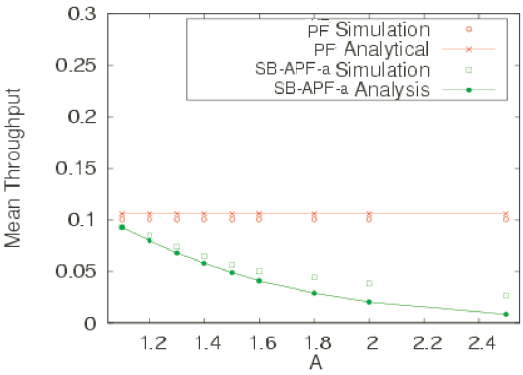

Fig. 13

MEAN THROUGHPUT OF LONG FLOWS

consistently overestimate the access probability for short flows and underestimate the mean throughput for long flows. This is once again due to the fact that the PF performance is biased against short flows as explained in section III. For that reason, it is better to separate short flows and long flows as we did in SB-HPF so that short flows profit fully from the preferential treatment they are given. As for performance, we see that we have an increase in the access probability for short flows at the expense of a decrease in throughput for long flows.

B. SB-APF-b

For this algorithm, aiming to increase the RICQ for short flows, we reduce the value of their average throughput by omitting to update it every time we serve a short-lived flow according to (8) but once every Y times. We hide from the PF algorithm the fact that these flows are being served as frequently as they really are, consequently, short flows will be served promptly and receive additional time slots.

1) Numerical Analysis: We now give a brief proof to show that updating the average throughput T_i of user i every $\frac{1}{Y}$ times will increase its probability to be served in comparison with a user j that is treated in a regular fashion according to the PF scheduling politic. Omitting to update T_i every time user i is served leads us to replace it by $T_i - a$ with $a < T_i$, the value of a increasing with the value of Y. For a fading Rayleigh channel, the conditional probability that user i has a relatively better channel than user j is:

$$\mathbb{P}(\frac{R_j}{T_j} \leq \frac{R_i}{(T_i - a)}|i) = 1 - e^{-x \cdot \frac{T_i}{(T_i - a)}}$$

The resulting asymptotic time fraction assignment P_i of user i is given by the following:

$$P_i = \int_0^{\infty} (1 - e^{-x \cdot \frac{T_i}{(T_i - a)}}) \cdot e^{-x} dx = \frac{T_i}{(2T_i - a)} \qquad (11)$$

From (11), we deduce that the asymptotic time fraction assignment P_j of user j is equal to $P_j = 1 - P_i = \frac{(T_i - a)}{(2T_i - a)}$. Given that $T_i > a$, it is obvious that $P_i > P_j$, and the greater the value taken by Y, the greater will be the value taken by a and consequently the more P_i is greater than P_j. We have proved then that SB-APF-b increases the chance of short flows to be served and thus reduces their transfer time. Unfortunately, a more elaborated analysis is not possible because we do not know what value will be taken by a for a given value of Y. Thus, we restricted to simulation results to evaluate the performance of SB-APF-b in comparison with PF. Due to lack of space and to the modest benefit obtained from this algorithm we refrain from presenting the results. The reason behind these performances is that short flows are often too short to profit from the preferential treatment they are given and, as they are not separated from long flows, they still endure the bias of PF.

VI. PF LAS

The third algorithm is a modified version of the flow size-aware scheduler LAS [8]. LAS favours short flows without prior knowledge of flow sizes. To this end, LAS gives service to the flow that has received the least service. An implementation of LAS needs knowing the amount of service so far received by each flow. LAS scheduling is optimal with respect to the average time in the system among all work-conserving disciplines that do not take advantage of precise knowledge of the flow length, when the service time distribution has a decreasing hazard rate (DHR) (which is the case for Internet flow traffic).

We calculate the mean response time $\overline{T}(\frac{S}{C})$ for flows with size S and average data rate C in LAS. Let $F(\frac{S}{C})$ be the distribution function of the flow service time. Let m_S^n be the n^{th} moment of the truncated distribution at S. Namely, $m_{S,C}^n = \int_{\frac{p}{C}}^{\frac{S}{C}} y^n dF(y) + (\frac{S}{C})^{\alpha} \cdot (1 - F(\frac{S}{C}))$.

The utilization factor for the truncated distribution is $\rho_{S,C} = \lambda \cdot m_{S,C}^1$ where λ is the intensity of arrivals at the cell (taken to follow a Poisson process).

The average response time of user i with service time S_i is:

$$\overline{T_i}(\frac{S_i}{C_i}) = (\overline{W}(S_i, C_i) + \frac{S_i}{C_i}) \cdot (1 - \rho_{S_i,C_i})^{-1} \qquad (12)$$

with $\overline{W}(S_i, C_i) = \frac{\lambda}{2} \cdot m_{S_i,C_i}^2 \cdot (1 - \rho_{S_i,C_i})^{-1}$

A. Wireless LAS

We run the LAS algorithm in the previous wireless environment with Rayleigh fading. We approximate the average sojourn time of user i with service time S_i from (12) (the average rate C_i being equal to $\mathbb{E}[R_i]$). Accordingly, the scheduler picks, at time slot t, user $i^* = arg_i max \frac{\mathbb{E}[R_i]}{S_i(t)}$.

B. PF-LAS

This algorithm seeks to improve the performance of wireless LAS by taking into account the instantaneous variations of the channel condition. Therefore, the scheduler picks, at time slot t, user $i^* = arg_i max\left(\frac{\mathbb{E}[R_i]}{S_i(t)} \cdot \frac{R_i(t)}{T_i(t)}\right)$, where $T_i(t)$ is the average throughput.

We know that $\frac{R_i(t)}{T_i(t)} = \frac{C_i \cdot x_i(t)}{C_i \cdot U_i(t)} = \frac{x_i(t)}{U_i(t)}$ and to get the average response time for user i, we replace in (12) S_i by $S_i \cdot \frac{U_i}{x_i}$. But as we modified the PF algorithm, the exponential throughput does not converge to what we obtained in (9) and consequently $U_i \neq \frac{g(n)}{n}$ where n is the number of active users. We compute then a lower bound on the average response time experienced by flows. A lower bound for a flow i is obtained when the latter realizes always the highest rate among the n possible rates, n being the mean number of users present in the system and when U_i keeps its initial value U_0:

$$\overline{T}_{max}\left(\frac{S_{i,max}}{C_i}\right) = \left(\overline{W}(S_{i,max}, C_i) + \frac{S_{i,max}}{C_i}\right)(1-\rho_{S_{i,max},C_i})^{-1}$$

with $S_{i,max} = \frac{S_i \cdot U_0}{\mathbb{E}[x_1,..,x_n]} = \frac{S_i \cdot U_0}{\sum_{k=1}^{n} \frac{1}{k}}$.

C. Numerical experiments

We consider a system where users initiate file transfer requests as a Poisson process of intensity $\lambda\pi\Re^2$ (traffic demand is uniformly distributed in the cell). Flow sizes are independent following the BP distribution $BP(1kbytes, 1000kbytes, 1.16)$. Users are served according to the Wireless LAS termed WLAS and according to our PF-LAS. In all experiments, We determine the average Sojourn Time for short flows and the average Throughput for long flows. We take $U_0 = 0.5$. Figure 14 displays the mean sojourn time as a function of flow size for short flows (whose size goes up to 40 Kbytes) for λ=20 indicating that the analytical formulae provide an exact estimate for WLAS and that the Lower Bound (LB) computed for the sojourn time in PF-LAS is tight enough that it can be fairly considered as an estimation of this sojourn time. As for results, the gain obtained in PF-LAS in terms of mean transfer delay is considerable.

Figure 15 displays the Mean Throughput of long flows for λ=20 and 15 and shows, contrary to the previous algorithms, how long flows profit greatly in PF-LAS and realize notably higher throughputs in comparison with WLAS.

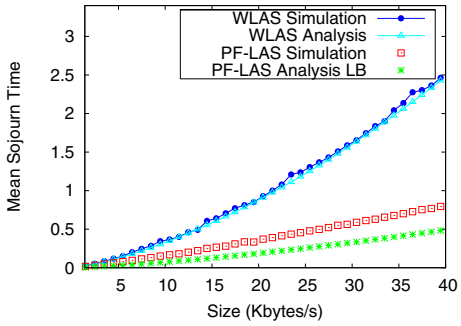

Fig. 14

MEAN SOJOURN TIME FOR $\lambda = 20.0$

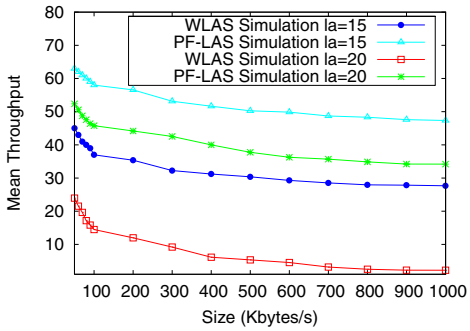

Fig. 15

MEAN THROUGHPUT FOR $\lambda = 15.0$ AND $\lambda = 20.0$

VII. CONCLUSION

This paper proposes and analyses three opportunistic scheduling approaches where flow size information is taken into account by the scheduling policy. The proposed size-aware schedulers reduce the latency for short flows while exploiting user diversity, thus, allowing the wireless channel to be utilised efficiently, without significantly disturbing the performances of long flows.

This work only validates the potential impact of the proposed scheduling mechanisms; a deeper analysis is being carried out, under more realistic scenarios, as well as an evaluation of the feasibility of the proposed solutions.

REFERENCES

[1] Bender P., Black P., Grob M., Padovani R., Viterbi A., CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic users, IEEE Communications Magazine, pp. 70-77, July 2000.

[2] Love R., Gosh A., Nikides R., Jalloul L., Cudac M., High-Speed Downlink Packet Access Performance, Proceedings of IEEE VTC Spring, 2001.

[3] Klein T. , Leung K. and Zheng H., Improved TCP Performance in Wireless IP Networks through Enhanced Opportunistic Scheduling Algorithms, Globecom 2004.

[4] Viterbi A.J., CDMA: Principles of Spread Spectrum Communication, Addison-Wesley, 1995.

[5] Bonald T., Proutière A., Wireless Downlink Data Channels: User Performance and Cell Dimensioning, MobiCom03, September 2003.

[6] Kushner H.J., Convergence of Proportional-Fair Sharing Algorithms Under General Conditions, IEEE Transactions on Wireless Communications, Vol.3, No.4, July 2004.

[7] Cohen J.W., The multiple phase service network with generalized processor sharing. Acta Informatica 12, 245-284.

[8] Rai I., Urvoy-Keller G., Analysis of LAS scheduling for Job Size Distributions with High Variance, SIGMETRICS03, June 2003.

[9] Crovella M., Bestavros A., Self-Similarity in World Wide Web traffic: Evidence and possible causes, IEEE/ACM Transactions on Networking, pp 835-846, December 1997.