Monitoring Proportional Fairness in cdma2000® High Data Rate Networks

Cédric Westphal

***Abstract*— As the 3G cdma2000® networks with High Data Rates get deployed, it is interesting to study the impact of the scheduling algorithm used in these networks, and to improve its performance. We consider here the setting where this scheduling algorithm is used in heterogenous channels, meaning channels for which the fading properties are not the same from one user to the next. Some users might encounter relatively steady conditions while other suffer from fast fading for instance. In heterogenous channels, it is known that the proportional fairness algorithm falls short of its goal to provide fair throughput to users that is proportional to their mean rate. We offer an easy way to implement improvements to this shortcoming. The idea for these improvements is to add an independent performance monitoring function to the proportional fairness algorithm that corrects unfairness should it appear.**

***Index Terms*— CDMA networks, Proportional Fairness, scheduling algorithm, forward link.**

I. Introduction

ONE of the drivers for the next generation of cellular networks is the expected need for higher bandwidth levels. For cdma2000® networks, a first step will be to have asymmetric bandwidths levels to and from the mobile device. The reason is that current traffic over the internet is indeed asymmetric: the most common application is web browsing, in which a client sends small requests to a web server and receives the larger web page content in return.

Similarly, in a cellular network, the forward channel from the base station to the mobile user, and the reverse channel, from the mobile user to the base station, exhibit very distinct properties. The forward channel can be controlled by a scheduler at the base station, and the base station acts as a central intelligence to regulate the traffic to the mobile users. The reverse channel, on the other hand, could be an asynchronous channel in which all users might contend for the use of the resource at the same time. In any case, the downlink [1] scheduling is a centralized problem, whereas the uplink scheduling is distributed over the mobile nodes.

The current asymmetry in the traffic thus corresponds to an asymmetry in the channel characteristics. HDR (high data rate) networks take advantage of this asymmetry. We focus in this paper in the scheduling algorithm for HDR networks. These networks are the one defined by the 3GPP2 standardization body in [1], and correspond to the next generation of cdma2000® networks [2], [3]. Note that this asymmetry is bound to disappear as the content becomes located in the mobile users, for instance for peer-to-peer file exchange, or interactive videophone streams.

In order to increase the capacity of the forward channel (the channel from the base station to the mobile device) in HDR, the scheduler at the base station allocates the whole capacity of the channel to one user at a time at the maximum power. This has been proved [4] to be optimal -in a more general setting than cdma2000® HDR-, as it cancels out the interference within one cell and does not increase the interference between cells.

This document focuses on the forward channel scheduling, and on the proportional fairness scheduling policy defined in [3]. We will describe in the next section the proportional fairness scheduling policy, and point to the literature of its flaws. In Section III, we present a solution to the problem of fairness in heterogenous channels. The idea is to define a metric that feeds back into the proportional fairness algorithm. It is a monitoring function that considers the input and output of the proportional fairness scheduler, and corrects trends that it deems unsatisfactory. In Section IV, we discuss a slightly different approach to the solution. Finally, as these proposals are early attempts at solving the problem, we sketch the future research directions we will pursue in Section V.

II. Proportional Fairness Fundamentals

A. Model

We consider a system in which N nodes compete for the channel. The channel is slotted. In real-life CDMA network, each slot is 1.66 ms in EV-DO (cdma2000® evolution Data Only) and 1.25 ms in EV-DV (evolution Data and Voice). To keep things general, we denote by τ the length of the time slot. A scheduler allocates traffic to one of the N nodes for the duration of each time slot. The scheduler receives as input a rate vector $\mathbf{R}^t = (r_1^t, \dots, r_N^t)$ which describes the achievable throughput r_i^t for user i during the next time slot t based on the channel conditions. We drop the superscript t whenever possible.

The rate vector $\mathbf{R}^t$ varies according to some stochastic processes in t. For simplicity here, we assume that the r_i^t are independent random variables extracted from N distributions $r_1, \dots, r_N$. The scheduler receives traffic for the N mobile nodes in N buffers. Each buffer size is infinite. We denote by $\eta^t \in \{1, \dots, N\}$ the queue picked by the scheduler to extract packets from at time t.

B. Proportional Fairness Policy

The proportional fairness (PF) policy computes some achieved rate using a sliding window average, and maximizes

C. Westphal is with the Communication System Laboratory, Nokia Research Center, Mountain View, California. E-mail: cedric.westphal@nokia.com

[1] Forward (respectively reverse) and downlink (respectively uplink) have equivalent meanings. They are used interchangeably in this paper.

the ratio of the achievable rate for the next time slot over this previously achieved rate. This allows nodes with poor channel condition to still gain access to the resource, as the less rate they receive, the higher their ratio. In our notations, the policy becomes:

$$\eta = \arg \max_{j \in \{1,\dots,N\}} \left(\frac{r_j}{R_j}\right) \quad \text{where} \tag{1}$$

$$R_j(t+1) = (1-\alpha)R_j(t) + \alpha \mathbf{1}_{\{\eta=j\}} r_j \quad \text{and } 0 < \alpha < 1 \tag{2}$$

The PF policy attempts to kill two birds with one stone:

- Optimize the throughput by taking into account in the choice of η the value r_j in the numerator, and
- Maintain fairness by keeping track of the achieved rate R_j in the denominator.

Both objectives are intimately combined together in the algorithm. This is what makes the PF algorithm very attractive in the homogenous case: it meets these two objectives in this simple and elegant manner.

C. Fairness in Heterogenous Channels

Many papers ([5] for instance, or [6] for a more general and theoretic result) have shown the unfairness of the PF algorithm. While this phenomenon has been thoroughly studied in these papers, we illustrate it here with a simple example.

We begin by plotting the throughput achieved by the Proportional Fairness Scheduling algorithm in the following set-up: 5 mobiles nodes are connected to a base station. All nodes request on average the same mean unit rate. However, the rates for nodes 1, 3 and 5 are subject to Rayleigh fading, whereas the nodes 2 and 4 are subject to a Gaussian fading. Since all means are equal, each users should receive 20 % of the bandwidth if the scheduling was fair and proportional. We consider Gaussian fading as it is an extreme case of the generally accepted Rician fading that illustrates nicely the behavior of the PF algorithm. Rician fading exhibits the same unfairness, but with slightly less difference between the nodes.

Fig. 1. Impact of variance on PF in Heterogeneous Channels

We varied the variance of the Gaussian from 4 % to 40 % with a 4 % step. We kept all the mean rates equal for all channels in doing so. The results are depicted in Figure 1. Each line corresponds to a value of the variance. The integer values on the x-axis correspond to the different channels. The line with the highest peak (at node 3) corresponds to 4% and the lines turn out to be ordered at their peak in the reverse order of the variance. The narrower the distribution of the Gaussian channel, the higher the peak at node 3, or the lower the throughput for node 2 and 4. Heterogeneity in the channels is of course the situation in real systems. It might not be Gaussian vs. Rayleigh. Yet, a base station path to the mobile node can be through line of sight, or through multipaths, the mobile station can be static or having different mobility patterns, the interferences can be local to one channel, etc, all of which will affect the channel characteristics.

III. A Monitoring Feedback to Proportional Fairness

Let's now turn our attention to how to fix the bias of PF with respect to heterogenous channels. We will present an evolution of the PF scheduling which address this unfairness.

We consider only scheduling policies that respect the current interfaces of the cdma2000® network: such an infrastructure is the result of a standardization consensus that is hard to modify. However, within these interfaces, scheduling is an implementation issue for the scheduler manufacturer. This can be phrased as:

Definition III.1: An admissible scheduling policy is a policy that bases its scheduling decision at time t only on the values $(r_1^s, \dots, r_N^s)$ for $s < t$.

Note that, as R_i belongs to the filtration generated by the rate vector $\mathbf{R}^s = (r_1^s, \dots, r_N^s), s < t$, the PF scheduling is obviously admissible. We introduce a scheduling policy that dissociates the monitoring goal from the throughput maximization goal, so as to achieve both satisfactorily. It uses the Proportional Fairness part for the latter goal, while it introduces a new feedback element for the former part.

A. Monitoring Proportional Fairness

Our solution to the heterogenous unfairness is to monitor the behavior of the scheduling algorithm, and to feed back this decision into the scheduling algorithm if some imbalance is detected. We describe this approach in this section. We present a very simple algorithm: the purpose is to illustrate the idea and demonstrate its effectiveness. Fine tuning the algorithm for optimal performance is the object of a subsequent document.

The quantities available to the scheduler are the requested rate vector $\mathbf{R}^t = (r_1^t, \dots, r_N^t)$ and the decision variables $R_1, \dots, R_N$.

The monitoring and feedback algorithm functions as follow (it is depicted in Figure 2 as well):

- The monitoring agent receives $\mathbf{R}$ at each time slot from the scheduler
- From $\mathbf{R}$, it computes an estimate $\hat{\mathbf{R}}$ of $E[\mathbf{R}]$ by:

$$\hat{r_i} = (1-\alpha)\hat{r_i} + \alpha r_i, \forall\, i \in \{1, \dots, N\} \tag{3}$$

- It receives the decision variables R_i at each time slot from the scheduler
- At every T units of time, it computes the ratio $\rho_i \triangleq R_i/\hat{r_i}$. For perfect fairness, this ratio should be the same for each node. It represents a form of gain achieved by the channel. $\hat{\mathbf{R}}$ is the mean rate that would be achieved by randomly allocating the channel, or by round-robin allocation. A perfectly fair allocation is not necessarily possible. For instance, a constant channel would have gain $\rho_i = 1$ no matter what the gain on the other channels is.
- It then measures the spread of this ratio. Several metrics can measure how much the ρ vector differs from a perfectly fair vector, that is, proportional to $(1, 1, \ldots, 1)$. We use: $\max(\rho_i)/\min(\rho_i)$.
- If there is too much discrepancy, for instance if $\max(\rho)/\min(\rho) > \chi$ for a preset threshold χ, then it resets the values of $R_1, \ldots, R_N$ in the scheduler by increasing the values R_j for j such that $\rho_j > \min(\rho) * \chi$. For these j, R_j is increased by a multiplier. By increasing these R_j, which PF uses for its decision, we make the channel j less attractive to the scheduler.

By increasing the value of R_j that stray to far from the other ones, the feedback makes it more difficult to pick the channel j. R_j does not necessarily represents the achieved mean rate anymore, as in traditional PF, but more some threshold for the node j to receive traffic. The update is made only every T units of time, and not at every time slot. This is to ensure that the proportional fairness algorithm has a chance to function properly, and to correct it only when necessary. In our simulations, we used $T = 60$ time slots. As we illustrate a proof of concept in this document, we only present values for T and χ for which the algorithm works. Finding out *optimal* values for T or χ, while an interesting exercise in control theory, is beyond the scope here.

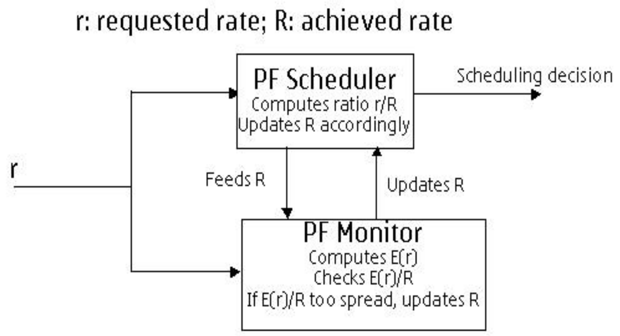

Fig. 2. Monitoring and Feedback for Proportional Fairness

B. Simulation Results

We simulated the the monitoring and feedback algorithm in a system with 14 nodes. All the channels had a Rayleigh fading channel with the mean rate fixed. The means were taken from [7], and spanned the range between 400 and 137 bandwidth units/s. The system makes a decision at each time slot. The simulations were ran for 14,000 time slots. The simulation tool used for this study here is matlab. We also assume all the channels and nodes have an infinite backlog of traffic. We used $\chi = 1.2$ and $T = 60$. All values R_j were updated per monitoring period. We used the same value as χ for the rate adjustment multiplier.

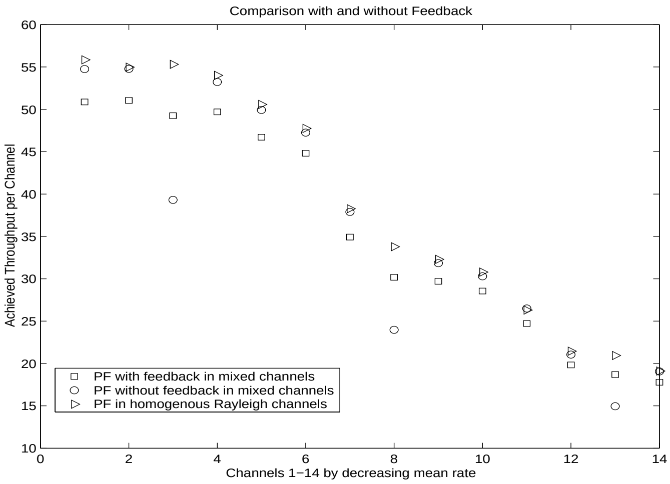

Fig. 3. Monitoring and Feedback Performance

Figure 3 depicts the simulation results in this set-up. We plotted 3 curves on this graph. As a reference, we plotted the throughput that proportional fairness would achieve in the homogeneous situation, with all Rayleigh fading channels. We then compared proportional fairness with and without feedback, in a heterogenous situation. Users 3,8 and 13 have the same mean rate as in the reference case, but with a Gaussian fading. Again, we use Gaussian as a middle point between a deterministic channel and a Rician channel. We encounter the behavior we described earlier in this document in the case without feedback: users 3,8 and 13 receive a much higher throughput than fairness would oblige. In the case with feedback, however, the system performance is in line with the reference case in terms of fairness, and close in terms of performance.

IV. Monitoring Allocated Throughput

The monitoring feedback we described in the previous section attempts to ensure more fairness from the PF policy. However, as we noted earlier, PF does itself attempt to jointly maintain fairness while maximizing the channel utilization. As such, there are redundant attempts here to maintain fairness. We thus propose to disable the fairness functionality from the PF policy, so as to delegate it solely to the monitoring feedback.

This means that the algorithm we consider does not update the value of the decision variables R_j if a node is *not* chosen, but let the monitoring algorithm tweak these values periodically. We denote this algorithm as the Proportional Allocated Rate (PAR) policy.

One aspect of the monitoring scheme described in Section III-A is that it is working as an enhancement of the built-in correcting mechanism of Proportional Fairness. A better way to add a correcting mechanism is to 'disable' the correcting function in Proportional Fairness. The idea is to have a scheduling policy that attempts at achieving an allocation that is proportional to the rate, and a fairness agent using the monitoring agent of section III-A. The scheduling policy is slightly different from proportional fairness. While we will detail the monitoring feedback later, the scheduling policy is as follows:

$$\eta = \arg \max_{j \in \{1,\ldots,N\}} \left(\frac{r_j}{R_j} \right) \quad \text{where} \tag{4}$$

$$R_j(t+1) = (1 - \mathbf{1}_{\{\eta=j\}}\alpha)R_j(t) + \alpha \mathbf{1}_{\{\eta=j\}} r_j \quad \text{and } 0 < \alpha < 1 \tag{5}$$

It can easily be seen that it is again equivalent to PF in homogenous channels, even without the feedback. However, in heterogenous channel, with different variance to mean ratios, we will see the ratio $R_j/E(r_j)$ tends to different stationary values for each channel: this makes the monitoring agent described in the previous situation for PF extremely efficient in this situation since the ratio $R_j/E(r_j)$ becomes an indicator of unfairness.

We study both elements now. First we consider the behavior of the scheduling policy defined in 5, and see how the ratio $R_j/E(r_j)$ evolves in the absence of the monitoring feedback. We will later see how we can use this quantity in the monitoring feedback mechanism to ensure fairness in the scheduling policy.

A. Analysis of Proportional Allocated Rate without Monitoring

First, we explain why the Proportional Allocated Rate without Monitoring allows the monitoring feedback to discriminate amongst unfairness. We study how this PAR policy performs in heterogenous situations.

To add some insight on the Proportional Allocated Rate in the heterogeneous situation, we consider a very simple set-up: two nodes are competing for the channel using the PAR scheduling algorithm described in (5). We use the same methodology and the same notations as in II. One node, that we denote node $\mathcal{R}$ is attached via a Rayleigh fading channel, and the other node, denoted by $\mathcal{C}$ has a constant rate. Also, assume that the update rule of the PAR algorithm (given in equation 5) is tilted so that the sliding average remembers only the last decision. This is the situation where $\alpha = 1$.

This set-up is a worst case situation for the PAR scheduling policy, both in term of the "adversary" and the choice of α, but is useful to give a good intuition of why the PAR policy turns out to highlight the different channel conditions, and assists the monitoring feedback in correcting the unfairness that might derive from the different channel conditions.

The decision rate of the PAR scheduling policy for the constant channel node $\mathcal{C}$ is always equal to 1 by construction. This means that the node $\mathcal{R}$ is picked only when its ratio $r(t)/R(t)$ is greater than 1. We drop the node index here as there is no confusion. It can be shown by induction that, with the assumption we made here,

$$R(t) = \max_{s=1,\ldots,t} (r(s), R(0)) \tag{6}$$

where $R(0)$ is the initial condition. This means that the decision rate $R(t)$ is an increasing function, and thus that the probability of the scheduler to pick node $\mathcal{R}$ decreases.

Further, for a given x, the probability to exceed x can be computed. It is equal to, if we define σ to be the parameter of the Rayleigh distribution:

$$\begin{aligned} P[R(t) \geq x] &= 1 - \prod_{s=1}^{t} P[r(s) < x] \\ &= 1 - (P[r < x])^t \\ &= 1 - (1 - P[r \geq x])^t \\ &= 1 - \left(1 - \exp(-\frac{x^2}{2\sigma^2})\right)^t \\ &\to 1 \text{ as } t \to \infty \end{aligned} \tag{7}$$

For any fixed x, the decision rate will be greater than x at some point with probability 1. After this (a.s. finite) time T, the probability that the scheduler picks $\mathcal{R}$ is less than that of having a random variable greater than x, namely:

$$P[\text{Scheduler picks } R \text{ for } t > T] \leq e^{-\frac{x^2}{2\sigma^2}} \to_{x \to \infty} 0 \tag{8}$$

The convergence rate can be estimated as well. We can compare for instance $R(t)$ with $\sqrt{\log(t)}$. For a fixed $\epsilon > 0$ and $\sigma = 1$ with no loss of generality,

$$\begin{aligned} P[\frac{R(t)}{\sqrt{\log(t)}} < \sqrt{1-\epsilon}] &= \left(1 - e^{(-(1-\epsilon)\log(t))}\right)^t \\ &= \left(1 - \frac{1}{t^{1-\epsilon}}\right)^t \\ &= \exp\left(t \log(1 - \frac{1}{t^{1-\epsilon}})\right) \\ &\sim \exp\left(-\frac{t}{t^{1-\epsilon}}\right) \\ &\sim \exp\left(-t^{\epsilon}\right) \\ &= \frac{1}{e^{t^{\epsilon}}} \end{aligned} \tag{9}$$

So, as

$$\sum_t P[\frac{R(t)}{\sqrt{\log(t)}} < \sqrt{1-\epsilon}] < \infty \tag{10}$$

and by Borel-Cantelli lemma,

$$P\left[\frac{R(t)}{\sqrt{\log(t)}} < \sqrt{1-\epsilon} \text{ infinitely often}\right] = 0 \tag{11}$$

This means that $R(t) \geq \sqrt{1-\epsilon}\sqrt{\log(t)}$ almost surely.

Thus the probability that the scheduler picks $\mathcal{R}$ after t units of time decreases at least as fast as $1/t^{\sigma^2}$. Note that the wider σ, the faster this probability decreases, and the less likely is $\mathcal{R}$ to be chosen.

This proves that, as time increases, the scheduler will stop picking the node $\mathcal{R}$ and only pick the node $\mathcal{C}$. The assumption we made can be relaxed: if the coefficient α is taken to be

$0 < \alpha < 1$, then the decision rate R is still an increasing function (as it is updated only if $r/R > 1$. It can be shown that it increases not as fast, but will still increase to ∞. The probability of picking $\mathcal{R}$ in this situation also decreases to 0 as well for the same reasons.

Remark IV.1: This behavior is corrected using the feedback agent, as described in the next section. The interest of the algorithm here is that it identifies clearly the unfairness created by the heterogeneity in the channel characteristics. Unlike PF, it does not make an attempt at correcting the unfairness: this allows for the monitoring agent to identify and to correct the unfairness easily. In Section III-A, we saw the monitoring agent could correct unfairness by monitoring the straight PF scheduling policy. However, it is sensitive to the choice of parameters, as it might conflict with the PF attempts at maintaining fairness. This sensitivity disappears when monitoring PAR scheduling.

Remark IV.2: The reasoning still holds if two nodes compete against each other so that one has a much smaller variance than the other, the means being equal. If there are more than two nodes, some being Rayleigh, other being constant, the rate of increase of each node's achieved rate R increases faster, as it is the one with the maximal r_j/R_j that it is picked. Basically, in this setup, the Rayleigh nodes compete to get out of the scheduling game first.

Remark IV.3: Of course, in real networks, the power is constrained to some limits. For some CDMA networks, the set of possible values is limited to a few discrete rates. However, the reasoning described above still holds: if one $\mathcal{R}$ node is connected through a Rayleigh fading channel and the other $\mathcal{C}$ node has a constant deterministic rate, then the $\mathcal{R}$ node will increase its decision rate value to the highest possible value in the set of possible values. After having reached this value R_{max}, the $\mathcal{R}$ node gets chosen with probability $\exp(-R_{max}^2/2\sigma^2)$. For a channel with mean say 300 kbps and a maximum possible value for instance 384 kbps, this probability is: 7.6%. The $\mathcal{C}$ channel gets chosen the other 92.4% of the time. Just to illustrate, a channel with a constant rate of 32 kbps will achieve a rate of 32 kbps 92.4% of the time: this is a 29.6 kbps average rate. The other channel achieves, after a transient period, a rate of 384 kbps 7.6 % of the time, namely an average of 29.2 kbps. This is neither proportional nor fair.

B. PAR policy with Monitoring Agent

As explained above, the behavior of the PAR policy allows for easy correction using the Monitoring Agent concept from Section III-A.

1) Monitoring Agent and Feedback Algorithm: The monitoring agent is exactly the same as for the proportional fairness scheduling algorithm. However, the PAR algorithm is designed so that the monitoring is made easier and more efficient. The PF algorithm attempts to build in a feedback within the value of the decision variable R_j by having it being a mean rate: if there is too much unfairness, and a node is not receiving any traffic, then the value R_j evolves so as to correct this. Of course, the structural limits of the PF algorithm prevents it from doing so. This is what the PAR algorithm combined with the monitoring corrects.

The monitoring and feedback algorithm functions as follow (it is depicted in Figure 2 as well):

- The monitoring agent receives $\mathbf{R}$ at each time slot from the scheduler
- From $\mathbf{R}$, it computes an estimate $\hat{\mathbf{R}}$ of $E[\mathbf{R}]$ by:

$$\hat{r_i} = (1-\alpha)\hat{r_i} + \alpha r_i, \forall\, i \in \{1,\dots,N\} \qquad (12)$$

- It receives the decision variables R_i at each time slot from the scheduler
- At every T units of time, it computes the ratio $\rho_i \stackrel{\Delta}{=} \hat{r_i}/R_i$
- It then measures the spread of this ratio: $\max(\rho_i)/\min(\rho_i)$
- If there is too much discrepancy, for instance if $\max(\rho)/\min(\rho) > \chi$ for a preset threshold χ, then it resets the values of $R_1,\dots,R_N$ in the scheduler by decreasing the values R_j for j such that $\rho_j < \max(\rho)/\chi$. For these j, R_j is reset so that $\hat{r_j}/R_j = \max_i(\rho_i)$.

By decreasing the value of R_j that stray to far from the other ones, the feedback makes it easier to pick the channel j. R_j does not necessarily represents the achieved mean rate anymore, but more some threshold for the node j to receive traffic. The update is made only every T units of time, and not at every time slot. This is to ensure that the proportional fairness algorithm has a chance to function properly, and to correct it only when necessary. In our simulations, we used $T = 500$ time slots, and updated only at most one R_j per update slot.

2) Simulation Results: We ran the monitoring and feedback algorithm on the set-up from 3 and 2. We used $\chi = 1.5$ and $T = 500$. We also decided, to simplify the simulation, to update only one value R_j per update period, as it did not impact adversely the performance of our monitoring scheme.

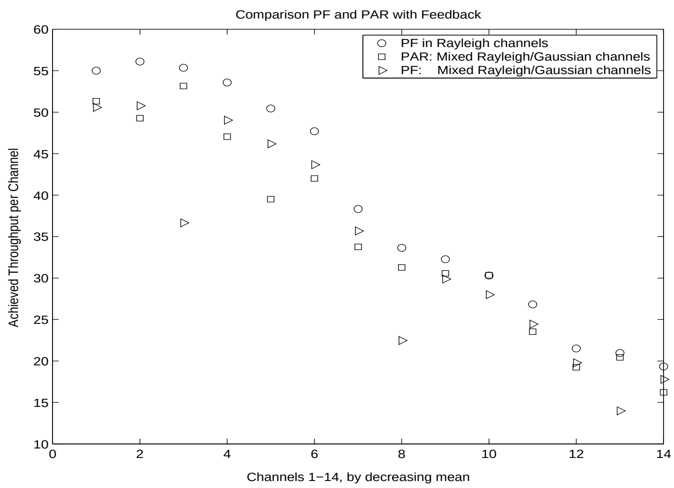

Fig. 4. Monitoring and Feedback Performance

Figure 4 depicts the simulation results in this set-up. We plotted 3 curves on this graph. As a reference, we plotted the throughput that proportional fairness would achieve in the homogeneous situation, with all Rayleigh fading channels. We then compared proportional fairness without feedback and PAR with feedback in an heterogenous situation: users 3,8 and 13 have the same mean rate as in the reference case, but with a

Gaussian fading. We encounter the behavior we described earlier in this document for PF: users 3,8 and 13 receive a much lower throughput than fairness would oblige. In the case of PAR with feedback, however, the system performance is in line with the reference case.

V. Conclusion

In this paper, we have studied the behavior of the cdma2000® proportional fairness scheduling algorithm in heterogenous channels. We have highlighted a propensity of this algorithm to behave unfairly to some users. The contribution of the paper is a simple feedback algorithm that corrects the flaw of the proportional fairness. We show also that, while this feedback algorithm works with the PF scheduling policy, it is easier to configure with a different scheduling policy, a slightly modified PF policy we denote the PAR policy. It works better with the PAR policy as this policy focuses on the throughput optimization and outsources the fairness management to the monitoring agent. By dissociating between fairness management and throughput maximization, the scheduling policy does not interfere with the monitoring agent.

The monitoring agent proves to offer significant improvement over the vanilla proportional fairness algorithm. Further, it requires changes only to the internal logic of the scheduler, but none to the other elements of the cdma2000® networks, and none to the interfaces between these elements and the scheduler.

References

[1] 3rd Generation Partnership Project 2 "3GPP2" *CDMA2000 High Rate Packet Data Air Interface Specification* 3GPP2 C.S0024 v. 4.0, October 2002.

[2] E. Esteves *The High Data Rate Evolution of the CDMA2000 Cellular System* Multiaccess, Mobility and Teletraffic for Wireless Communications. Vol. 5, pp.61-72, Kluwer Academic Publishers, 2000.

[3] P. Bender, P. Black, M. Brob, R. Padovani, N. Sindhushayana, A. Viterbi *CDMA/HDR: a Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users* IEEE Communications Magazine, July 2000, pp.70-7.

[4] A. Bedekar, S.C. Borst, K. Ramanan, P.A. Whiting, E.M. Yeh, *Downlink scheduling in CDMA data networks*, Proc. IEEE GLOBECOM'99, pp. 2653-2657, 1999. or extended version: Technical Report PNA-R9910, CWI.

[5] Holtzman, J.M. *Asymptotic analysis of proportional fair algorithm.* Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on , Volume: 2 , 30 Sept.-3 Oct. 2001 Page(s): F-33 -F-37 vol.2

[6] S. Borst, *User-level Performance of Channel-Aware Scheduling Algorithms in Wireless Data Networks* Proceedings of INFOCOM 2003, San Francisco

[7] S. Shakkotai, A. Stolyar *Scheduling for multiple flows sharing a time-varying channel: the exponential rule* American Mathematical Society Translations, Series 2, A volume in memory of F. Karpelevich, Yu. M. Suhov, Editor, Vol. 207, 2002.

[8] M. Andrews, K. Kumaran, L. Ramanan, A.L. Stolyar, R. Vijayakumar, P. Whiting, *CDMA data QoS scheduling on the forward link with variable channel conditions* Bell Labs Technical Report, April 2000.