A CHANNEL PREDICTIVE PROPORTIONAL FAIR SCHEDULING ALGORITHM

Hans Jørgen Bang	*Torbjörn Ekman*	*David Gesbert*
Dept. of Informatics University of Oslo h.j.bang@fys.uio.no	UniK - University Graduate Center Kjeller, Norway torbjorn@unik.no	Eurecom Institute Sophia-Antipolis, France gesbert@eurecom.fr

ABSTRACT

There is growing interest in the area of cross-layer design. This paper addresses the problem of multi-user diversity scheduling together with channel prediction. Recent work demonstrates that the mobile radio channel can be linearly predicted, quite accurately, several milliseconds ahead in time for realistic Doppler spreads. We show how this information can be exploited in an original fashion to improve the so-called throughput-fairness trade-off in multi-user wireless fading channel scheduling. Simulations show that the proposed algorithm can give considerable throughput gains without compromising fairness. The current approach builds on the proportional fair scheduling technique but can also be generalized to other criteria.

1. INTRODUCTION

There has been considerable interest recently in so-called cross-layer design techniques. In the area of user resource allocation for wireless fading channels, the gain of channel-dependent scheduling exploiting multi-user diversity was clearly demonstrated [1]. In this approach, channel access is granted to the user with the best channel quality among the cell users (max SNR or max capacity strategy). To cope for the case of unequal fading statistics or average SNRs, it was later proposed to exploit fairness-aware criteria which offer a compromise between cell throughput and equity between the users. This is exemplified among others in the proportional fair algorithm [2, 3]. In the particular case of proportional fair scheduling (PFS), a time constant parameter can be chosen to specify over what time period fairness (in terms of realized throughputs) between the users should be maintained.

In parallel, recent work on channel modeling and prediction has shown the feasibility of prediction of Rayleigh fading channels over horizons of up to 0.25 wavelengths with reasonable accuracy [4, 5, 6]. In this paper we propose to combine channel prediction with resource allocation with the aim of improving the system capacity. Although intuitively appealing, this problem has received little attention so far. In [7] scheduling together with prediction was addressed from the point of view of prioritization of traffic flow with different QoS classes.

Here we address a single antenna system although the proposed concepts can be generalized to the multi-antenna case. We make the following key points:

1. There exists a fundamental trade-off between total cell throughput and fairness in wireless multi-user scheduling. Schemes allowing to push the trade-off (by improving one metric while not degrading the other) are therefore of great interest.

2. For system scenarios where fairness is to be maintained over long periods of time (as compared to the coherence time of the fading), capacity maximization is obtained through a normalized max SNR strategy. Thus channel prediction is of minor interest since we tend to give access to the user with the best *current* channel compared to the average channel quality.

3. For scenarios where tighter fairness constraints are used, there is a significant gain to be obtained from a channel prediction-aware scheduler.

In the paper, we give both qualitative and quantitative explanation to this phenomenon. More specifically we propose a generalization of the PFS algorithm capable of exploiting fading prediction. The generalization is based on the optimization of a specific multi-user utility function. We propose both an optimal and a sub-optimal (low complexity) greedy multi-user scheduling algorithm for the optimization. We also propose the use of the so-called Jain's fairness index [8] as a tool for measuring the fairness within the context of wireless scheduling. Jain's fairness index has been used to measure the performance of networking protocols, but is little known within the wireless communication theory community. We demonstrate how the proposed algorithm can improve the throughput-fairness trade-off while being robust with respect to decaying prediction quality (for increased prediction horizons). The algorithm is evaluated through extensive Monte Carlo simulations and intuitive results are provided.

2. SYSTEM MODEL

We consider the downlink of a single interference-free cell with N simultaneously active users served by one base station (BS). The scheduling is organized on a slot by slot basis, i.e. one and only one user is served during any given slot. The scheduler resides at the BS and decides prior to each slot which user the BS shall transmit data to. We use $i^*(k)$ to denote the user scheduled in slot k.

Our key assumption is that estimates of the users' supported data rates for the current and $L-1$ future slots are available to the scheduler. The supported rate for the ith user in slot $k+l$, as predicted in slot k, will be denoted $\hat{R}_i(k+l|k)$. $R_i(k)$ will be shorthand for $\hat{R}_i(k|k)$.

3. PROPORTIONAL FAIR SCHEDULING

For the PFS algorithm the user scheduled in time slot k is given as

$$i^*(k) = \arg\max_{i=1,.,N} \frac{R_i(k)}{T_i(k)}, \tag{1}$$

where $T_i(k)$ is the ith user's average throughput in a past window. The average throughputs are updated each time slot according to

$$T_i(k{+}1) = \begin{cases} \left(1-\frac{1}{t_c}\right)T_i(k) + \frac{1}{t_c}R_i(k) & i = i^*(k) \\ \left(1-\frac{1}{t_c}\right)T_i(k) & i \neq i^*(k) \end{cases} \tag{2}$$

where t_c is a time constant adjusted to maintain fairness over a pre-determined time horizon.

Alternatively the PFS algorithm can be defined in terms of the system utility function

$$U(k) = \sum_{j=1}^{N} \log T_j(k), \tag{3}$$

where $\log T_j(k)$ should be interpreted as the level of "satisfaction" or utility for user j. Let $T_j(k{+}1|i)$ denote $T_j(k{+}1)$ given that user i is scheduled in time slot k, then (1) is equivalent to

$$i^*(k) = \arg\max_{i=1,.,N} U(k+1|i), \tag{4}$$

where

$$U(k+1|i) = \sum_{j=1}^{N} \log T_j(k+1|i).$$

A proof can be found in the Appendix. Thus, the user who gives the largest instantaneous reward in the system utility function $U(k)$ is scheduled in each time slot.

4. PREDICTIVE SCHEDULING

In this section we extend the PFS algorithm to a predictive scheduling scenario. The basic idea will be to use both current and future rate estimates to maximize the long-term average value of the system utility function $U(k)$.

Let $\hat{U}\big(k+L|(i_1, i_2 \ldots i_L)\big)$ denote the estimated value of $U(k{+}L)$ in time slot k, given that user i_l is served in time slot $k{+}l{-}1$. The scheduling vector $\mathbf{i}^*(k) = \big(i^*(k), i^*(k+1), \ldots, i^*(k{+}L{-}1)\big)$ in time slot k that maximizes $U(k{+}L)$ is obtained according to

$$\mathbf{i}^*(k) = \arg\max_{\mathbf{i}\in\mathcal{F}} \hat{U}(k+L|\mathbf{i}), \tag{5}$$

where $\mathcal{F} = \{1, 2 \ldots N\}^L$ is the set of feasible scheduling combinations. Through a full search the scheduling is optimized over a block of L time slots at a time, instead of only one time slot as in (4). Note that for $L = 1$ this reduces to the standard PFS algorithm.

The main problem with the scheduling strategy outlined above is that we rely on estimates of the future supported rates to compute $\hat{U}(k+L|\mathbf{i})$. Reliable short range predictors for the received power have been proposed [4, 5, 6] and these predictors can be used to predict the supported data rates. However these predictions degrade rapidly for longer prediction ranges. To fix the schedule for a long block might result in large errors. A more robust approach is to redo the scheduling for each time slot k and only effectuate the first component of the scheduling vector.

For each time slot k:

1. Update the prediction of the supported rates, $\hat{R}_i(k+l|k)$.

2. Compute a scheduling vector

$$\mathbf{i}^*(k) = \big(i_1^*(k), i_2^*(k), \ldots, i_L^*(k)\big),$$

that suggests to which users' time slots k to $k+L-1$ should be allocated.

3. Schedule the user given by the first component of $\mathbf{i}^*(k)$ in time slot k, i.e. $i^*(k) = i_1^*(k)$. The other components of $\mathbf{i}^*(k)$ serve as auxiliary variables.

Another drawback of the algorithm is the complexity of the full search for the scheduling vector. A low complexity algorithm that renders good but possibly suboptimal results is presented in the following.

4.1. An iterative algorithm for obtaining $\mathbf{i}^*(k)$

As opposed to doing a full search for the optimal scheduling vector we propose a coordinate ascent-type algorithm. At each iteration one component of the scheduling vector is updated with the other components held fixed. In order to describe the algorithm we use the following notation.

- Let $\mathbf{i}^n(k) = \big(i_1^n(k), \ldots, i_L^n(k),\big)$ denote the computed scheduling vector after n iterations.

- For an arbitrary vector $\mathbf{i} = \big(i_1, \ldots, i_L\big)$, let $\mathbf{i} \overset{l}{\leftarrow} i$ denote the vector $\big(i_1, \ldots, i_{l-1}, i, i_{l+1}, \ldots, i_L\big)$.

The coordinate ascent algorithm can now be described as follows.

1. To initialize the algorithm let

$$\mathbf{i}^0(k) = \big(i_2^*(k-1), i_3^*(k-1), \ldots, i_L^*(k-1), 1\big).$$

Note that the last component of $\mathbf{i}^0(k)$ is set to 1. This is merely for convenience and will not affect later iterations.

2. At each subsequent iteration we recompute one component of the scheduling vector. The $(n+1)$th iteration is given by

$$\mathbf{i}^{n+1}(k) = \mathbf{i}^n(k) \overset{l}{\leftarrow} i_l^{n+1}(k),$$

where $l = L - (n \bmod L)$ and

$$i_l^{n+1}(k) = \arg \max_{i=1,.,N} \hat{U}\big(k + L|\mathbf{i}^n(k) \overset{l}{\leftarrow} i\big). \quad (6)$$

3. When $\mathbf{i}^n(k) = \mathbf{i}^{n-L}(k)$ we will have $\mathbf{i}^m(k) = \mathbf{i}^n(k)$ for all $m \geq n$ and we have converged to a solution.

We can see from **2.** that at each iteration we can only obtain a better solution in the sense that

$$\hat{U}\big(k + L|\mathbf{i}^{n+1}(k)\big) \geq \hat{U}\big(k + L|\mathbf{i}^n(k)\big).$$

Hence the algorithm will necessarily converge to a maximum. Furthermore, fast convergence can be expected as only limited amounts of new channel state information is introduced at each time step and the initial scheduling vector is based on a scheduling vector rendering a maximum in the previous time step. We next state a result that indicates that the computational complexity for each iteration can be significantly reduced.

Lemma: Let

$$\hat{T}_i\big(k + L|(i_1, \ldots, i_L)\big) = (1 - \frac{1}{t_c})^L T_i(k) + \sum_{l=1}^{L} \frac{1}{t_c}(1 - \frac{1}{t_c})^{L-l} \hat{R}_i(k + l - 1|k)\delta(i - i_l), \quad (7)$$

where $\delta(\cdot)$ is the Kronecker delta function, $T_i(k)$ is given by (2) and $\hat{R}_i(k + l - 1|k)$ is a prediction of the rate of the ith user $l-1$ time slots ahead. Then (6) is equivalent to

$$i_l^{n+1}(k) = \arg \max_{i=1,.,N} \frac{\hat{R}_i(k + l - 1|k)}{\hat{T}_i(k + L|\mathbf{i}^n(k) \overset{l}{\leftarrow} 0)}. \quad (8)$$

Proof: See Appendix.

Thus it is not necessary to explicitly evaluate the function $\hat{U}\big(k+L|\mathbf{i}^n(k) \overset{l}{\leftarrow} i\big)$, for each user i, as suggested in (6).

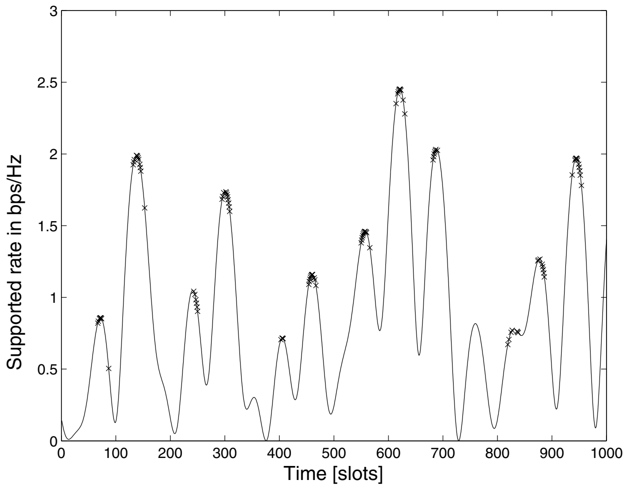
5. SIMULATIONS

In this section, we evaluate the performance of the proposed algorithm through simulations. We consider the case where all users have infinite backlogs. The simulation results are obtained for Rayleigh fading channels with time correlations given by Jake's model. The average SNR is 0 dB and the time slot Doppler frequency product is 0.01 for each user. This means that the terminals move one wavelength in 100 time slots. We use the Shannon Capacity to estimate the supported data rates.

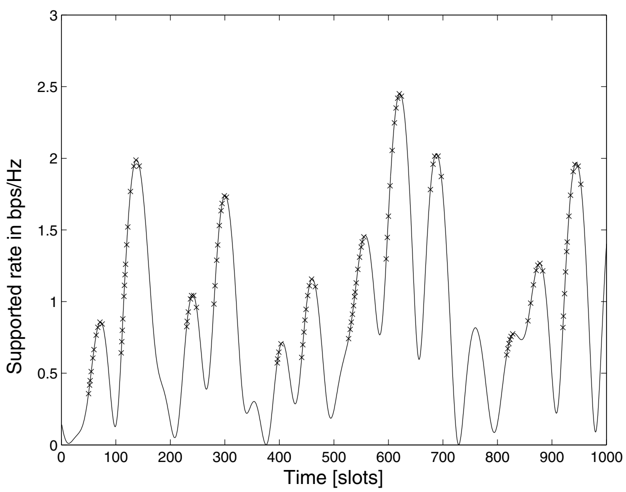
In order to generate realistic estimates of the users' future supported rates we use a linear FIR MMSE predictor with 128 coefficients to predict the future complex fading gains from past noisy observations of the channel. The channel power gain used in the Shannon Capacity is obtained as the absolute square of the predicted complex fading gain. The quality of the prediction will depend on the Channel to Estimation Error Ratio which is set to 20 dB. The NMSE for the complex fading gain prediction ten time slots ahead is roughly 10^{-2} but for one step ahead it is only 10^{-3}. Thus, the error in the estimated rate for one time slot ahead can be disregarded.

We first demonstrate the performance of the iterative predictive algorithm introduced in Section 4.1 in a qualitative manner. Consider a scenario with 10 users, $t_c = 100$ and $L = 21$. Figure 1(a) shows a snapshot of the supported rates for one user, where a superimposed cross corresponds to an allocated slot. Note that all allocated slots are tightly clustered around local fading peaks. By comparing with Figure 1(b), which shows the same scenario with the standard PFS algorithm, we can conclude that there appears to be significant increase in performance by using prediction.

We next try to validate these claims in a more rigorous manner. We now consider a system with 15 users. Figure 2 shows the system throughput for the standard and the predictive PFS algorithm as a function of the parameter t_c. The prediction horizon $L - 1$ is set to 10 slots for the predictive algorithm. On average 20 iterations were required to obtain convergence. It can be seen that there is an increase in throughput with the predictive algorithm for all values of t_c. Note however, that the largest gains (20%) occur for smaller t_c values. This is intuitive because for large t_c both algorithms approaches the max SNR scheduler which takes neither the past nor the future into account.



(a) Predictive PFS, $t_c = 100$ and $L = 21$.



(b) Standard PFS, $t_c = 100$.

Fig. 1. Snapshot of supported rates for one user. A superimposed cross corresponds to an allocated slot with (a) the predictive (b) the standard PFS algorithm. The predictive algorithm leads to higher throughputs since the allocated slots are more tightly clustered around the fading peaks.

To quantify the degree of fairness over shorter time intervals we use Jain's fairness index [8], which is defined as

$$J = \frac{\left(\sum_{i=1}^{N} \mathcal{T}_i\right)^2}{N \sum_{i=1}^{N} \mathcal{T}_i^2},$$

where $\mathcal{T}_i$ is user i's average throughput (computed over a rectangular window of size W slots[1]). Jain's fairness index measures the spread in the users' average throughputs $\mathcal{T}_i$ and will always be in the range $1/N$ to 1. It can easily be verified that $J = 1$ indicates absolute fairness and $J = 1/N$ indicates no fairness, i.e all resources are allocated to one single user.

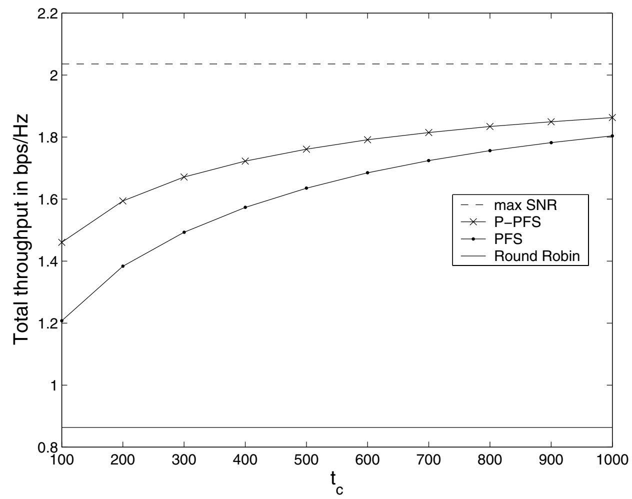


Fig. 2. Total throughput as function of t_c for the predictive and the standard PFS algorithm. The prediction horizon $L-1$ is set to 10 slots for the predictive algorithm.

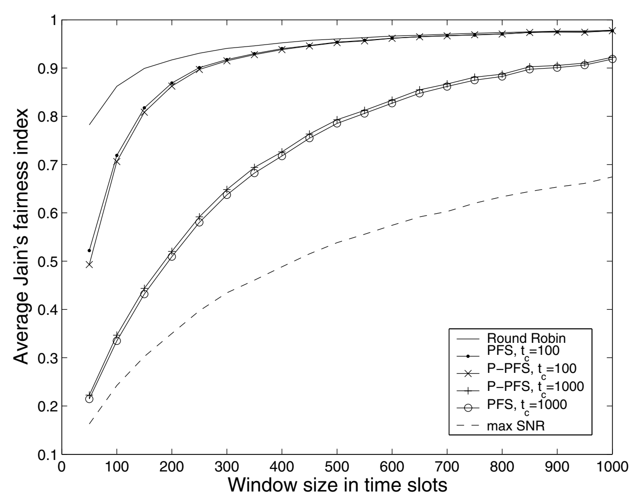


Fig. 3. Average Jain's fairness index as function of window size for the predictive and the standard PFS algorithm with $t_c = 100, 1000$.

Figure 3 shows Jain's fairness index, averaged over multiple windows, as a function of window size for the standard and predictive PFS algorithm with $t_c = 100, 1000$. Observe that the level of fairness is virtually the same for the two algorithms. There is a negligible reduction in fairness for

[1]Note that by adjusting the window size W the time horizon that fairness is computed over is adjusted accordingly.

$t_c = 100$ and a slight increase in fairness for $t_c = 1000$ with the predictive algorithm.

Figure 3 also shows the corresponding plots for the Round Robin and max SNR algorithm. Not unexpectedly there is a large penalty in fairness with the Max SNR algorithm. One should also note that even though the Round Robin algorithm is perfectly fair when allocating the radio resource in time, it is not perfectly fair when considering the actual throughputs over a finite window.

6. CONCLUSION

In this paper we have introduced a wireless scheduling algorithm capable of exploiting fading predictions. At a reasonable increase in complexity and without compromising fairness the total throughput was significantly increased compared to the standard PFS algorithm.

7. APPENDIX

PROOF OF EQUIVALENCE BETWEEN (6) AND (8)

We first note that $\hat{T}_i\big(k+L|(i_1,\ldots,i_L)\big)$ as defined in (7), equals the estimated value of $T_i(k+L)$ in time slot k based on the predicted rates and that user i_l is served in time slot $k+l-1$. This can easily be verified by solving (2) as a difference equation. We can therefor write

$$\hat{U}\big(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} i\big) = \sum_{j=1}^{N} \log \hat{T}_j(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} i). \quad (9)$$

Observe next that

$$\hat{T}_j(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} i) = \hat{T}_j(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} 0)$$

for $j \neq i$ and

$$\hat{T}_j(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} i) =$$
$$\hat{T}_j(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} 0) + \frac{1}{t_c}(1-\frac{1}{t_c})^{L-l}\hat{R}_j(k+l-1|k)$$

for $j = i$. Equation (9) can therefore be written as

$$\hat{U}\big(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} i\big) = \sum_{j=1}^{N} \log \hat{T}_j\big(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} 0\big)$$
$$+ \log\left(1 + \frac{1}{t_c}(1-\frac{1}{t_c})^{L-l}\frac{\hat{R}_i(k+l-1|k)}{\hat{T}_i\big(k+L|\mathbf{i}^n(k)\stackrel{l}{\leftarrow} 0\big)}\right).$$

This completes the proof since only the last term in the expression above depends on i and $\log(\cdot)$ is a monotone increasing function. We finally note that for $L=1$ and $l=1$ we obtain the equivalence between (1) and (4) as a special case.

8. REFERENCES

[1] R. Knopp and P. Humblet, "Information capacity and power control in single-cell multiuser communication," in *Proc. of Int. Conf. on Communication*, June 1995, vol. 1, pp. 331–335.

[2] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.

[3] P. Viswanath, D.N.C. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Transactions on Information Theory*, vol. 48, pp. 1277–1294, June 2002.

[4] A. Duel-Hallen, S. Hu, and H. Hallen, "Long-range prediction of fading signals," *IEEE Signal Processing Magazine*, vol. 17, pp. 62–75, May 2000.

[5] R. Vaughan, P. Teal, and R. Raich, "Short-term mobile channel prediction using discrete scatterer propagation model and subspace signal processing algorithms," in *IEEE VTC2000-Fall*, Boston, MA, Sept. 2000.

[6] T. Ekman, *Prediction of mobile radio channels, modeling and design*, Ph.D. thesis, Uppsala University, Sweden, 2002.

[7] N.C Ericsson, *Revenue maximization in resource allocation: Applictions in wireless communication networks*, Ph.D. thesis, Uppsala University, Sweden, 2004.

[8] R. Jain, *The art of computer systems performance analysis*, John Wiley and Sons, 1991.