On the Impact of User Mobility on Call Admission Control in WCDMA Systems

Leonardo Badia, Michele Zorzi and Alessandro Gazzini – Università di Ferrara

***Abstract*—In this paper, we present a study of Admission Control in 3G systems. In particular, the behavior of algorithms already presented in the literature is analyzed, with respect to their implementation in UMTS-like systems, and a model of trade-off between the QoS metrics, blocking and dropping probability, is presented. Obtained performance is discussed and analyzed under different points of view, and important related aspects are highlighted. Known algorithms are evaluated, in terms of fairness and generality of the performance when realistic models for mobility, data rate and discontinuous transmission (DTX) are taken into account. Finally, better models and possible ways to optimize these aspects are proposed.**

I. Introduction

Third Generation Mobile Communication Systems are based on Code Division Multiple Access (CDMA) multiplexing and access techniques. CDMA systems allow an improvement of system capacity with respect to Frequency or Time Division Multiple Access systems, by avoiding the problems of channel allocation, being theoretically possible to manage all users with the same channel. This phenomenon is called *soft capacity* of a CDMA system, as opposed to the *hard capacity* of FDMA and TDMA systems, in which the maximum number of users is fixed by the amount of physical resources.

On the other hand, code-based multiplexing is limited by Quality of Service (QoS) requirements, essentially depending on the power levels between each user and the base station (BS) the mobile user is connected to. In order to have an acceptable power level at the receiver for each connection, a power control (PC) mechanism is implemented in CDMA systems, with the goal to appropriately tune the transmitted powers.

Even with PC, the system performance is however limited: in general, the admission of users to the system is always possible, at the price of a general performance degradation for all active users. It is possible that a set of users with their QoS objectives does not admit a feasible solution, and in this case the PC algorithm diverges. Thus, in this case, a previously admitted user must be dropped in order to guarantee the requested service to other active users. Of course, from the users' point-of-view, cutting off an existing call is an event that should be avoided.

Henceforth, the access of users to the system must be controlled, or congestion may arise and cause call dropping. This means that a check should be done to forecast if the system is near congestion, and in this case the new call requests should be refused. Of course this control must not be too conservative, because blocking a new call is still an undesirable event (though less annoying than dropping an already existing call).

This work has been supported by Ericcson Research.

Call admission control (CAC) can be performed by following iterative real-time procedures [2] [5] [6], or by heuristic algorithms, that operate a threshold comparison [1] [3] [4] [7] [8] [9] [10].

Following [1] and [5], we can speak of *Feasibility based CAC* (*F-CAC*) for the first class of algorithms, and divide the second class into *Number-* or *Interference-based CAC* (*N-CAC* and *I-CAC* respectively). This further division is based on the heuristic used (number of user vs. some measure of the total power).

The first approach leads to analytically correct solutions, that however require long evaluation times, mainly because of the duration of the *testing phase* and other problems connected with computational complexity. Moreover, algorithms of this kind present good accuracy, but also problems under the aspect of *fairness*, i.e., these algorithms act well if used to maximize the performance from the server's side, but their adaptation to requests of users' satisfaction is more difficult. For these reasons, no clear decision has been made as to how to select and implement these schemes in practical systems.

The second approach is more interesting from the point-of-view of a real implementation: in these "instantaneous" algorithms the evaluation time is much shorter, because the capacity of the system is approximated by using a heuristic based on measuring a quantity, that is simpler to obtain than the general description of the system, while of course being less accurate.

Lower computational complexity and better performance lead us to emphasize the role of CAC algorithms based on heuristics, especially for the case of power measurements (I-CAC). Then our goal is to analyze, especially for this class of algorithms, the performance in terms of blocking and dropping probability, and discuss parameter optimization.

Our research analyzes the obtained performance under different aspects (i.e., cases of trade-off, fairness, statistical properties), with considerations that allow to identify new ways to improve the system.

Moreover, the major contribution of this work is a proposed approach to CAC that is aware of mobility differences among users, which can be easily tracked by the BSs. The simulations performed have shown that traditional approaches lead to unfairness if users with different mobility patterns coexist in the same system, and that a Mobility-aware Interference-based

CAC (MICAC) can provide much better fairness performance.

Finally, another contribution of this work is the possibility to extend the results previously found to distinguishing metrics other than mobility, such as data rate or activity factor of the users, which can also lead to unfairness of the algorithms. It is shown that techniques proposed to take into account users' mobility can be applied successfully in this case as well.

This work is organized as follows: in Section II we study the performance of Interference based CAC. Furthermore, Sections III-VI explain the feature of Mobility aware-CAC and offer a comparison between different approach by which users' mobility can be managed. Finally Section VII concludes the paper.

II. Global Performance of I-CAC

Simulations have been performed with a simulator of the UMTS system, in which some user dynamics have been implemented.

The simulation environment presents a deployment of the users based on a structure of 3×3 hexagonal cells wrapped onto itself so as to have no "border effect" [1].

In radio channel propagation, in addition to the path loss, both fast fading and shadowing have been taken into account: fast fading with the well known multi-oscillator Jakes' simulator [13] and shadowing with Gudmunson's correlation model [14]. Doppler frequency for the Jakes' simulator has been set equal to $f_c + (v/\lambda)$ hertz, where v is mobile speed, $\lambda = f/c$ the wavelength, that is equal to 0.16 in the simulator and f_c is a constant term equal to 2 Hz [2]. The parameter of the log-normal distribution of the shadowing is $\sigma = 4dB$.

We consider users whose speed is re-determined every 0.1 s, so that the amplitude is Gaussian distributed, with assigned mean and variance, and the direction is turned of a random angle uniformly distributed between -0.25π and $+0.25\pi$.

Users are generated, or already connected users terminate their call, following a birth–death Poisson process. When a new user arrival time is calculated, even its mobility parameters are randomly determined, by assigning it to one of 4 mobility classes with equal probability. In practice, we have indeed 4 Poisson processes, which correspond to 4 kinds of user: stationary (mean speed $v_m = 0.0$m/s, with standard deviation $v_s = 0.0$ m/s), slow ($v_m = 4.0$m/s, $v_s = 0.5$ m/s), medium ($v_m = 8.0$m/s, $v_s = 1.0$ m/s), fast ($v_m = 12.0$m/s, $v_s = 1.0$ m/s). When users are admitted the Power Control tries to guarantee a minimum SIR $\gamma^{tar} = 4.5$dB.

The examined algorithm for Admission Control is RPCAC [3] with different threshold values. It uses the received power as heuristic for the admission, i.e., the new call is admitted if the BS receives from other mobiles a power value that is under a given threshold. Thus, different values of the threshold P_{th} in the RPCAC algorithm are different received power levels. In our simulations they are normalized to the average power contribution that a MS, $0.5d$ away from the BS (d is cell radius), gives to P_{tot} when it transmits at maximum power level.

Another curve, introduced only for the sake of comparison, is called *Admit all* and corresponds to giving access to the network to every user that requests it (i.e., $I_{th} = \infty$). So, no calls are blocked and only the probability of dropping can be calculated.

The studied metrics are the probability of blocking a generic user that requests to be admitted (P_b), the probability of being dropped for a user already in the system, due to overload of the network (P_d), and a weighted combination of these two metrics, i.e., $P_b + 10P_d$, being call dropping generally considered more annoying than call blocking during the admission phase. These metrics are evaluated as a function of the mean load of each cell in the network, expressed in erlang/cell.

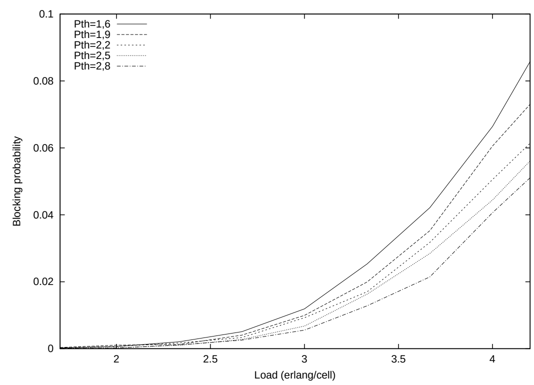

Fig. 1. Block probability for RPCAC

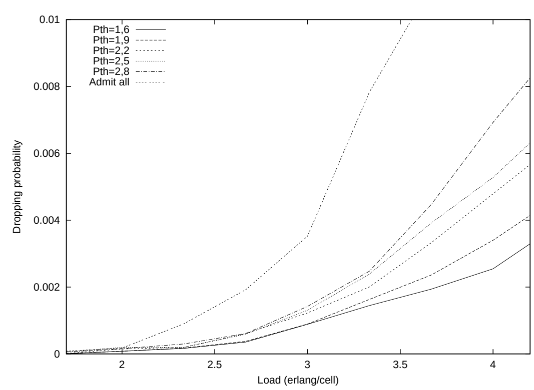

Fig. 2. Drop probability for RPCAC

Let us consider Figures 1–3: they represent P_b, P_d and the linear combination $P_b + 10P_d$, showing their behavior for different threshold values.

As it can be observed, when P_{th} for RPCAC varies, blocking and dropping probability present an interesting trade-off, because if the threshold is decreased, i.e., the system becomes more conservative, P_b increases whereas P_d decreases. How-

[1] Note that this wrapped structure implies 9 replies of the cells, so the number of simulated cells is higher. Moreover, the consistence of the results also for cases of 4×4 and 5×5 cells has been verified.

[2] This additional constant term allows to take into account the environment mobility, i.e., to assign a non-zero Doppler frequency even to stationary users with $v = 0$.

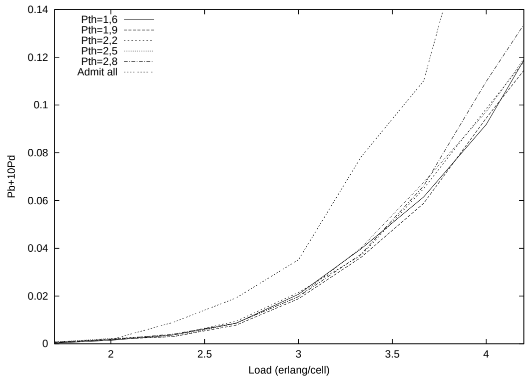

Fig. 3. $P_b + 10P_d$ for RPCAC

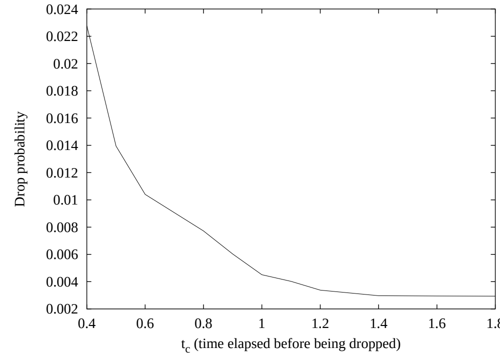

Fig. 4. P_d vs. t_c with fixed $1/\mu$ = 20s

ever, Figures 3 shows that the linear combination $P_b + 10P_d$ is almost independent of the chosen threshold.

It can be concluded that the choice of the threshold does not affect the global performance (the linear combination has almost the same values in both RPCAC and HCCAC cases), but implies great variability of the point in which the trade-off between blocking and dropping is cut. Moreover, Figures 1–3 show that a power-based Admission Control allows a significant improvement with respect to the "Admit All" situation (i.e., call blocking is performed in a smart way, so that a great number of dropping events is avoided).

The results shown in this section have been obtained by using a specific approach to evaluate when a user is to be dropped (approaches of such kind are called in the following *drop policies*). Other simulations have shown that the results for the performance metric P_d are strongly dependent on the drop policy we use, thus it can be useful to investigate which policy obtains better performance. Moreover, deeper investigations show that only the mean value of P_d is often not sufficient to correctly estimate the grade of service for the users: in fact, it is possible to obtain low mean values even with a rough policy, that on the other hand may be *unfair*. This undesirable effect heavily affects especially systems in which users are strongly different, e.g., in terms of their mobility parameters. In fact, unfairness becomes more evident if we consider details about different mobility classes of the users, in particular we put in evidence the effect of the drop policy, as will be done in following Sections.

III. Continuous Time under Threshold policy

The drop policies can be studied by dividing them in two steps: the detection of the congestion, and the choice of the dropped user. In our simulations, trace is kept of every user's SIR, and if the SIR of a user remains below a threshold γ^{th} (i.e., $\gamma_i < \gamma^{th}$) for a specified amount of time, say t_c, congestion is detected and that user is dropped. This policy is called in the following *Continuous Time under Threshold* (CTuT) policy.

In this case, performance is good if only mean values are considered, but the dropping process is unfair and highly dependent on users' parameters like mobility and call duration time.

For example, consider Figure 4, where effect on the performance of one of the two main parameters of CTuT policy, the evaluation time t_c, is considered.

Different values of t_c are considered, whereas γ^{th} and $1/\mu$ have been set to $\gamma^{tar} - 0.5$ dB and 20 s respectively, where γ^{tar} is the target SIR of the PC. Figure 4 shows that t_c must be conservatively estimated (e.g., a good choice could be 1.2 s) or a performance degradation occurs. However, too high an evaluation time could imply a slow reaction for the system, and this could be undesirable if the traffic in the network is high.

Moreover, this global strategy does not obtain good results under the aspect of fairness: in fact, for the mobility classes defined in Section II, it can be shown that dropped users are part of the first mobility class (fixed users) in more than 50% of the cases. Figure 5 shows this situation in the left chart, whereas the one to the right is a similar situation with 6 mobility classes instead of 4. Here the 6 average values of the speed are: $0,\ 2,\ 5,\ 12,\ 15,\ 20\,\mathrm{m/s}$ and the standard deviations are: $0\,\mathrm{m/s}$ for stationary users, $0.5\,\mathrm{m/s}$ for pedestrian ones and $1\ \mathrm{m/s}$ for vehicular ones.

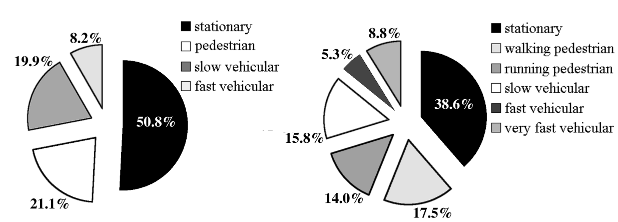

Fig. 5. CTuT: distribution of dropped users vs. mobility classes

IV. Users' removing strategies

A way to improve fairness could be to change the way users are removed. We can implement a different users' removing strategy rather than removing the worst one (i.e., the one that has been identified in congestion). Two simple possibilities are to drop a user at random or to drop the last entered user. This second choice could be intuitively explained as follows: since before the last admission the system was not congested, the last admitted user could often be identified as the cause of congestion.

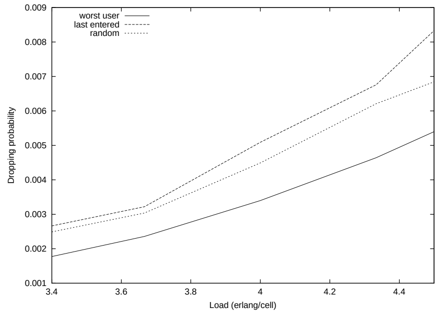

Fig. 6. CTuT: comparison users' removing strategies

Figure 6 shows that in order to obtain good performance, a congestion detection mechanism should always be associated with the removal of the worst user. If other policies are used, i.e., the last entered user or a randomly chosen is dropped, higher fairness is obtained, but at the price of a higher dropping probability. In particular, both policies obtain a perfectly fair distribution of dropped users among mobility classes (since even the last entered user has the same probability to belong to any class), and the blocking probability remains the same, being the only variation in the drop policy. However, there is an increase of the dropping probability, that is higher when the last entered user is dropped (thus this policy appears to be unnecessary, as it obtains worse performance than a random user removing policy).

However, the real problem is in the congestion detection part: the differences between the users lead unavoidably to a trade-off between fairness and goodness of the average performance. The problem of the CTuT policy is that it is not tunable in order to cut the trade-off appropriately. In fact, if it is supposed that the parameters γ^{th} and t_c are fixed, the only way to improve the performance is to change thresholds in order to give access more frequently to fixed users, or mobile users. Note that there is a trade-off between stationary users, which achieve worst performance as they cause an increase of dropping probability, and mobile users, which guarantee a lower dropping probability, but cause a decrease of the fairness, because they are dropped by the system less frequently.

This fact can be highlighted by seeing Figure 7, where our proposed multi-threshold Admission Control has been considered. In this case, it is supposed that the system tracks the speed of the mobile users, that belong to six mobility classes as described before. When a new user sends a request for admission to the system, the RPCAC algorithm uses a variable admission threshold for received power, depending on the user's speed that the system has previously tracked. In our simulations, there are 4 threshold, as described in Table I. They

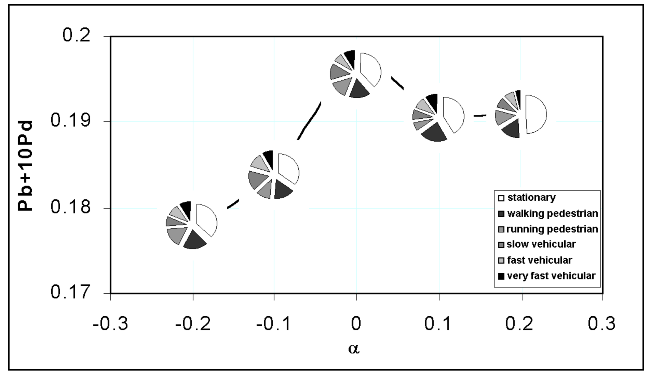

Fig. 7. CTuT: distribution of dropped users' mobility classes vs. system performance

are intended for fixed, slow, average speed and fast users, respectively. Table I also shows how the thresholds are changed if adjustments are made: for the sake of simplicity, it can be done by changing a single parameter α.

Threshold name	speed range (in m/s)	value
Z_0	< 0.1	$1.9(1+5\alpha)$
Z_s	$0.1 \div 5$	$1.9(1+2\alpha)$
Z_a	$5 \div 12$	$1.9(1-2\alpha)$
Z_f	> 12	$1.9(1-5\alpha)$

TABLE I
MULTITHRESHOLD VALUES AS FUNCTION OF α

If α is increased, vehicular users are accepted more frequently than fixed or walking pedestrians. This improves the performance, as the dropping probability is reduced: the blocking probability is increased, although this is a side-effect of the reduced dropping probability, and the trade-off between blocking and dropping is cut at a point of higher average QoS. However, the fairness of the system is further decreased, because the percentage of dropped users belonging to the stationary or pedestrian classes are not significantly decreased, whereas these users are blocked more frequently. On the other hand, if α is decreased, it becomes easier for a slow user to be accepted. This causes however the percentage of dropped users belonging to the stationary or pedestrian classes to be highly increased. Figure 7 has been constructed as follows: every chart on the figure represent the percentage of dropped users of each class, for a given value of the α-parameter. Moreover, the line between the charts and the position of the charts themselves are proportional to the performance metric $P_b + 10P_d$ defined in Section II. So, Figure 7 shows that for $\alpha = 0.2$ almost half of the dropped users are fixed and more than 75% are either fixed or pedestrian. That implies that the probability of being dropped for fixed users is more than 5 times greater than for vehicular users, and so in this case the system can not be considered fair. We can conclude that both ways to operate lead to poor performance, being the intrinsic fairness of CTuT policy extremely low.

An interesting observation that can be made is that the global performance of the system when the threshold are equal for all mobility classes (i.e., $\alpha = 0.0$) is poorer than the cases in which different kinds of users are admitted in a different way. This fact is a consequence of a general feature of Admission Control, that could be better performed when there is a priority order between the users (in this case, when $\alpha > 0$ fixed users are priorized with respect to mobile users, and vice versa).

A conclusion can be extracted from this remark: in a system where there are different mobility classes, a small variation of the thresholds could be suitable, if it is able to decrease the dropping probability without significantly increasing the probability of blocking the most penalized class. However, a larger variation of the threshold may perturbe the system too much, so it is undesirable. Thus, in the analyzed case, choices like $\alpha = \pm 0.1$ seems to be both better performant than the single threshold case $\alpha = 0$: here, the obvious disadvantage of having different access for each mobility class is overruled by the dropping probability decrease.

V. Positive/Negative Reward

Other congestion detection policies can be identified, in order to guarantee greater fairness than the CTuT policy. A similar policy that seems to be more accurate than CTuT is a Positive/Negative Reward policy (PNR), in which a counter is set to 0 for each user, and a call is dropped when the user counter goes below a given threshold. The counter update happens at the end of each frame, and it is such that a user whose SIR is below a threshold loses 1 point, whereas when the SIR exceed the threshold the counter is increased by k points (usually $k > 1$): in this way drop events are less frequent. Moreover, note that CTuT can be considered as a particular PNR policy in which $k = 0$.

However, this policy is unfortunately even more unfair than the CTuT. In fact a larger number of stationary users is dropped, since almost always a fixed users located where the coverage is bad has a very high probability to have its counter decreased for all frames, while a mobile user can gain positive rewards as it moves or as it has a frame with high SIR due to fast fading.

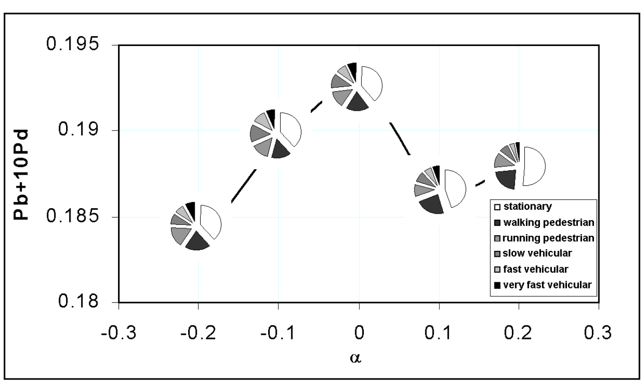

Fig. 8. PNR: distribution of dropped users' mobility classes vs. system performance

Results for $k = 5$ are showed in Figure 8, where the threshold is the same of the CTuT policy and the users are dropped when their counters reach -120. Note that this case is comparable with the situation depicted in Figure 7, because the algorithm acts almost identically to the CTuT for stationary users (which receive burst of negative rewards when they are in bad coverage), while it advantages the mobile users when they have a SIR over the threshold, even if it happens only for a frame.

The parameter α on the x-axis has the same meaning as in Figure 7, i.e., is a way to change the thresholds in order to guarantee easier access to certain classes. In the single threshold case ($\alpha = 0$) the PNR policy improves the global performance (the metric $P_b + 10P_d$ is lower) but at price of having a more unfair system. Moreover, in cases with $\alpha \neq 0$ we have more unfairness and poorer performance than in the CTuT cases. We can conclude that this policies is useful to lower the drop probability but is useless to have a fairness improvement.

VI. Drop policies by average SIR

Another class of heuristics that is indeed different from CTuT consists of policies in which the SIR is averaged over a given time interval. An example of drop policy that uses this criterion is the *Mean SIR under Threshold* (MSuT) policy. Whereas CTuT allows better treatment to high speed mobile users, MSuT tends to advantage fixed users, because the mean values are worse when the Doppler frequency is higher.

However, the MSuT has a problem connected with the power control, that is implemented by changing the transmitted power by ± 0.5dB, with the goal of keep the SIR over γ^{tar}. This implies that negative peaks in the SIR (that are common for users which are near to a newly established call) are not compensated in the mean value, since the power control does not generate peaks, but tries to force the convergence to an assigned value. Then, when the SIR rises over the threshold γ^{tar} the PC stops the increase of the transmitted power, and if it happens after a negative peak, even short in time, the average SIR is kept low for a long time.

The performance of MSuT policy is not shown, since it is quite similar to the case in which CTuT is used with a user removing strategy that drops the last entered users. This means that, independent of fairness, the global performance is too low to be acceptable.

We propose a way to extend the MSuT policy so that it has both good performance and fairness and call it *Mean SIR in 2 out of 3 frames under Threshold* (MS2/3uT). Here, in order to avoid problems with negative peaks, the average SIR of each user is computed frame-by-frame, and the values of the *three* latest frames are stored. At each frame, the lowest value is discarded, and the two remaining values are averaged. If the resulting value is below a given threshold γ^{th}, the user is dropped.

This mechanism has several advantages: first of all, it is fast to compute, since it requires only evaluation of three frames. Moreover, it does not depend on the call duration time, which is usually longer than three frames. Finally, if the request of new calls can be considered a rare event (i.e., λ is not too high, so that the probability of having two accesses in two adjacent frames is close to zero), it allows to neglect the negative peaks due to the new calls in computing the SIR, because the frame in which the new call arrives will be discarded.

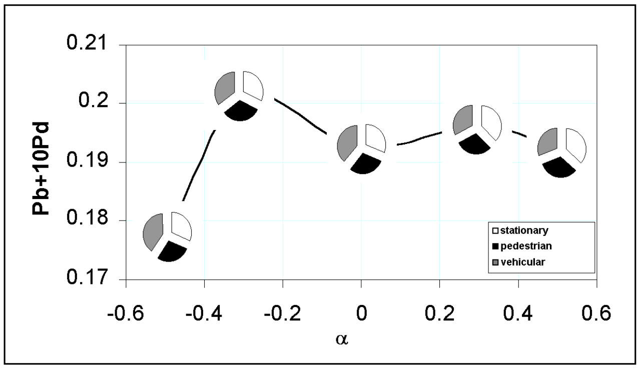

Fig. 9. MS2/3uT: distribution of dropped users' mobility classes vs. system performance

Simulations for MS2/3uT have been performed with the following parameters: instead of the six mobility classes of the CTuT, only three values for the average speed have been chosen: in this scenario, mobile users with average speed of 0 m/s, 6 m/s, 12 m/s have been considered, that are called stationary, slow and fast respectively. Admission threshold for the pedestrian class is always $Z_p = 1.9$, while thresholds for stationary and vehicular users are $Z_0 = 1.9(1 + 0.5\alpha)$ and $Z_v = 1.9(1 - 0.5\alpha)$ respectively. The threshold γ^{th} has been let equal to $\gamma^{tar} - 3$ dB. Under all other aspects, Figure 9 is identical to Figures 7 and 8. For better reference, the percentages depicted in each chart are also represented in Table II. It is clear that MS2/3uT policy has a higher degree of fairness with respect to CTuT. In fact, even without tuning the α-parameter, the three percentages are similar and the system can be considered fair. If a wider fairness is desired, Table II shows that a choice like $\alpha = -0.3$ could address this request, although Figure 9 shows a corresponding performance degradation.

α	%stationary	%pedestrian	%vehicular
-0.5	0.3134	0.2798	0.4068
-0.3	0.3208	0.3225	0.3567
0	0.3138	0.2997	0.3865
0.3	0.3714	0.3001	0.3285
0.5	0.3657	0.3245	0.3098

TABLE II

DROPPED USERS VS. MOBILITY CLASSES AS FUNCTION OF α

As a final remark, it can be concluded that such a policy is surely more suitable under the aspect of fairness, but obviously presents a small degradation (even if it can be managed by appropriate threshold choices) in the performance metric values, and this is due mainly to a higher number of dropped calls (the blocking probability is essentially the same as when CTuT is used). Finally, a policy similar to MS2/3uT, with a generalization of the parameters could be optimal to be used as drop policy for the UMTS system, in order to obtain good fairness without compromising the performance.

VII. CONCLUSIONS

In this paper, Call Admission Control instantaneous threshold algorithms are modeled and discussed, and their application to Third Generation systems has been emphasized.

Simulation results show that Mobility aware Interference based Call Admission Control allows to improve the performance when an instantaneous heuristic evaluation of the network is implemented. In fact, simulations have shown that, even though the results may be globally good, improvement under the aspect of fairness may still be necessary. Different systems have been implemented and simulation results have been discussed under the aspects of sensitivity to their parameters and fairness.

The way to optimize the performance can be identified in a better model of the drop policy, in which not only the global blocking and dropping probability, but even the behavior with respect to each mobility class is considered.

REFERENCES

[1] Y. Ishikawa, N. Umeda, "Capacity design and performance of call admission control in cellular CDMA systems", *IEEE Journal on Selected Areas in Communications*, Volume 15 Issue 8, pp 1627–1635, October 1997.

[2] M. Andersin, Z. Rosberg, J. Zander, "Soft and safe admission control in cellular networks", *IEEE/ACM Transactions on Networking*, Volume 5, Issue 2, pp 255–265, April 1997.

[3] C.-Y. Huang, R.D. Yates, "Call admission in power controlled CDMA systems", *Vehicular Technology Conference, 1996: Mobile Technology for the Human Race*, IEEE 46th, Volume 3, pp 1665–1669, 1996.

[4] Z. Liu, M. El Zarki, "SIR-based call admission control for DS-CDMA cellular systems", *IEEE Journal on Selected Areas in Communications*, Volume 12, Issue 4 , pp 638–644, May 1994.

[5] L. Nuaymi, P. Godlewski, C. Mihailescu, "Call admission control algorithm for cellular CDMA systems based on best achievable performance", *Proc. Vehicular Technology Conference, 2000.* VTC 2000 - Spring. IEEE 51st, Volume 1, pp 375–379, 2000.

[6] D. Kim, "Efficient interactive call admission control in power-controlled mobile systems", *IEEE Transactions on Vehicular Technology*, Volume 49, Issue 3, pp 1017–1028, May 2000.

[7] J. Kuri, P. Mermelstein, "Call admission on the uplink of a CDMA system based on total received power", *Proc. ICC'99*, Volume 3, pp 1431 -1436, June 1999.

[8] Y. Ma, J.J. Han, K.S. Trivedi, "Call admission control for reducing dropped calls in code division multiple access (CDMA) cellular systems", *Proc. INFOCOM 2000.* Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, pp 1481–1490, 2000.

[9] H. Holma, J. Laakso, "Uplink Admission Control and Soft Capacity with MUD in CDMA", *Vehicular Technology Conference, 1999.* VTC 1999 - Fall. IEEE VTS 50th, Volume 1, pp 431–435, 1999.

[10] J. Outes, L. Nielsen, K. Pedersen, P. Mogensen, "Multi-cell admission control for UMTS", *IEEE Vehicular Technology Conference May 2001*, IEEE VTS 53rd, Volume 2, pp 987–991, May 2001.

[11] C.-Y. Huang, R.D. Yates, "Rates of convergence for minimum power assignment algorithms in cellular radio systems", *Baltzer/ACM Wireless Networks*, April 1998.

[12] K. Kim, Y. Han, "A call admission control with thresholds for multi-rate traffic in CDMA systems" *Proc. Vehicular Technology Conference, 2000*, IEEE 51st, Volume 2, pp 830–834, Spring 2000

[13] W. C. Jakes, *Microwave Mobile Communications* - New Jersey, IEEE Press, 1993

[14] G. Stüber, *Principles of mobile communication* Boston, Kluwer, 1996