Mobility-aware admission control schemes in the downlink of third generation wireless systems

S-E. Elayoubi, T. Chahed and G. Hébuterne
GET/Institut National des Télécommunications
{salah_chedine.elayoubi, tijani.chahed, gerard.hebuterne}@int-evry.fr

Abstract—In this paper, we develop novel CAC algorithms that take into account the effect of mobility of users both inside and outside the cell in the downlink of 3G mobile systems. We first study the system capacity including the other-cell interference, subject to feedback between cells. We then obtain effective bandwidth expressions for calls as a function of both their location in the cell as well as their class of traffic (voice vs. data). We next use this formulation to derive two mobility-aware admission control algorithms: i. a Priority CAC, where calls are accepted not only upon resource availability, but also through acceptance ratios that reflect their levels of priority, ii. a Squeezing CAC, where elastic calls may be squeezed to a minimum agreed value, giving way to admit more calls in the system and to secure further ongoing mobile users. Using Markovian analysis, we obtain several performance measures, namely the blocking probability, the dropping probability, both intra and inter-cell, as well as the overall cell throughput. We eventually investigate the performance of our CAC and show how to extend the Erlang capacity bounds, i.e., the set of arrival rates such that the corresponding blocking/dropping probabilities are kept below predetermined thresholds.

I. INTRODUCTION

Third generation mobile communication networks need to support a variety of multimedia services. The system capacity has then to be increased significantly. To do so, Wideband Code Division Multiple Access (WCDMA) has been chosen as the radio access technique. The system is then interference-limited and a robust Connection Admission Control (CAC) is needed to guarantee Quality of Service (QoS).

Several CDMA-oriented CAC algorithms have been developed, considering the Signal-to-Interference Ratio (SIR) as the determinant parameter in accepting or not a new call; the idea being mainly that a new call is accepted if its contribution to the overall interference at the base station does not make the latter exceed a given value [14][16], or alternatively if the new call does not make the SIR of an ongoing user fall below a target value [9][11][21]. However, these works did not compute exact values for the blocking probabilities, and hence cannot be used for an exact dimensioning of the system.

Yet, other works studied the capacity of CDMA systems using Markov chains, and obtained analytical values for the blocking probabilities, but they focused solely on the uplink [2][18]; it is well known, however, that traffic in 3G systems is asymmetric, with the major part of data traffic supported by the downlink. The work in [15] computed exact blocking rates in the downlink, but failed to consider the maximal transmission power of the base station as an additional CAC constraint.

Moreover, these works considered a static CDMA system, where the QoS of a user does not depend on its mobility in the cell. This assumption could be somehow acceptable in the uplink, where a perfect power control makes the SIR homogeneous in the cell until the maximal transmission power is reached [11]. This is not true in the downlink where the SIR is largely dependent on mobility. In fact, the level of interference at the mobile station in the downlink will depend largely on its location in the cell: the farther is the mobile terminal from the base station, the higher is the interference it experiences from adjacent cells [11]. This factor, i.e., distance $r$ between the base station and the mobile terminal, plays a discriminating role between different users in the downlink.

Moreover, SIR depends on the other-cell interference that is subject to feedback between nearby base stations: the other-cell interference felt by the mobiles in the target cell makes their SIR decrease, the base station must then increase its transmission power to insure the required QoS. This, in turn, affects the other-cell interference in the adjacent cells, and so on. This feedback was studied in the uplink [26], but, in the downlink, an exact calculation of the other-cell interference is not a straightforward task, as it depends on the position of the user in the cell. We solve this problem by a quantization of the system. The idea is to decompose the cell into a finite number of concentric circles, so-called rings, and to define an effective bandwidth expression relative to each ring.

Note that the notion of rings appears, not only in 3G networks, but also in other new wireless technologies offering high data rates, like in the HSDPA (High Speed Downlink Packet Access). In these systems, the base station transmits at full power to only one user in each time slot and only a discrete set of achievable peak rates is possible, which defines a corresponding set of concentric rings in the cell [5]. The main task in HDR is to design efficient and fair scheduling policies, as a simple admission control independent of user locations will be sufficient [5].

We hence develop in this work an admission control algorithm, based on a realistic model of the radio interface, that handles the mobility issue in the downlink of 3G wireless systems and implements priorities between different classes of multimedia calls. Our contributions are:

- Accounting for feedback in the other-cell interference.
- An analytical modeling using Markov analysis that yields closed form expression of the blocking probability.
- Accounting for mobility within the cell and between adjacent ones. We obtained closed form expressions for...
intra and inter-cell dropping probabilities.

- Derive acceptance ratios in a first proactive CAC algorithm that gives priorities to calls depending on both their class and location. We show that priorities help increase the Erlang capacity of the system.
- Quantify, within a second reactive CAC algorithm, the effect of squeezing (downgrading) data traffic on the overall performance, which is still largely unknown because of the trade off between data rates and call durations.

The remainder of this paper is organized as follows. In section II, we calculate the other-cell interference and develop an effective bandwidth formulation for different classes of users. In Section III, we present our first Priority CAC algorithm and analyze it by Markov chains. The performance measures, i.e. blocking and dropping probabilities as well as cell throughput, are also calculated. Our Squeezing CAC algorithm is presented in Section IV. In Section V we validate our analytical model through simulations and address the Erlang capacity issue. Section VI eventually concludes the paper.

II. Model

A. SIR and power equations

We consider a homogeneous DS/CDMA cellular system with hexagonal cells, of radius $R$, uniformly deployed and numbered from 0 to $\infty$. Users belong to $S$ classes of multimedia traffic, with $K^x_r$ active users of class $s$ in cell $x$. The main notations used in this section are detailed in Table I.

In the downlink, the SIR of mobile $(x, c, m)$ is [11]:

$$\gamma_{c,m} = \frac{P^x_{c,m}}{\sigma^2 q^y_{x,c,m} + \frac{r^y_{x,c,m}}{N} + \frac{1}{N} \sum_{i=1}^S K^x_i P^x_{i,k}}$$

(1)

where $P^x_{c,m}$ is the power emitted by base station $x$ for mobile $(x, c, m)$. $N$ is the spreading factor and $\epsilon \in [0, 1]$ is the orthogonality factor due to multipath propagation affecting all signals, including target one. $\epsilon = 1$ if there is no multipath propagation, and is equal to 0 in a complete multipath environment; it varies usually between 0.4 and 0.6. $\sigma^2$ is the power of the Gaussian noise and $I^y_{x,c,m}$ is the other-cell interference experienced by user $(x, c, m)$. $q^y_{x,c,m}$ is the path loss between base station $y$ and mobile $(x, c, m)$ [1]:

$$q^y_{x,c,m} = L \times (r^y_{x,c,m})^\alpha (S_{x,c,m}/10)$$

(2)

$r^y_{x,c,m}$ being the distance between base station $y$ and mobile $(x, c, m)$ and $S_{x,c,m}$ is a Gaussian random variable due to shadowing, with zero mean and standard deviation equal to $\varsigma$.

Let $\gamma_c$ be the target SIR for insuring the desired QoS for class-$c$ users. The power emitted by base station $x$ is

$$P^x = \frac{\sum_{c=1}^S \gamma_c \sum_{m=1}^{K^x_c} (\sigma^2 q^y_{x,c,m} + r^y_{x,c,m})}{1 - \frac{1}{N} \sum_{c=1}^S K^x_c \gamma_c}$$

(3)

However, the other-cell interference depends on the powers of all base stations:

$$I^c_{c,m} = \sum_{y \neq x} P^y_{c,m} q^y_{x,c,m}$$

(4)

Eqns. (3) and (4) indicate that the calculation of the downlink other-cell interference shall be iterative, as the other-cell interference at each mobile station depends on the powers emitted by all other-base stations and vice-versa.

B. Cell decomposition

The downlink is limited by the maximum transmission power of the base station $W^B$ [11][13], and so: $P^x \leq W^B$.

Then, the CAC equation for cell $x$ is:

$$\sum_{c=1}^S K^x_c \gamma_c \left[1 - \epsilon \frac{W^B}{N^B} + \sigma^2 q^y_{x,c,m} + I^c_{c,m} \right] \leq W^B$$

(5)

This CAC equation is more constraining than classical ones where a call is accepted if it does not degrade the SIR of ongoing users [15][21], or if it does not make the system reach its pole capacity [2][3], i.e., the number of users that makes the power go to infinity, as the maximal transmission power is finite and the pole capacity can thus never be reached.

Let $f(r, \xi) = (1-\epsilon) W^B + \sigma^2 q^y_{x,c,m} + I^c_{c,m}$ denote the interference coefficient. The first term is due to the intra-cell interference, the second term to the Gaussian noise. And the third term, $I^c_{c,m}$, is generated by the other-cell interference.

Let us now investigate this interference coefficient if we do not consider the feedback between cells. In this case, if we assume that surrounding base stations emit with power $P^B$, the moments of the other-cell interference are given by [11] :

$$E[I^c_{c,m}] = P^B e^{b(\xi)} \sum_{y \neq x}(r^y_{x,c,m})^2 \left[1 - Q\left(\frac{10\log_{10}(r^y_{x,c,m})}{2\varsigma} - \sqrt{2}\omega\Omega\right)\right]$$

$$E[I^c_{c,m}^2] \leq P^B \Omega \frac{1}{2} (r^y_{x,c,m})^2 \left[1 - Q\left(\frac{10\log_{10}(r^y_{x,c,m})}{2\varsigma} - \sqrt{2}\omega\Omega\right)\right]$$

+ 2 \sum_{y \neq x}(r^y_{x,c,m})^2 \left[1 - Q\left(\frac{10\log_{10}(r^y_{x,c,m})}{2\varsigma} - \sqrt{2}\omega\Omega\right)\right]$$

where $\Omega = \ln(10)/10$, $b = \sqrt{2}/2$ and $Q(.)$ is the complementary cumulative distribution function of a standard (zero mean, unit variance) Gaussian process.
The downlink other-cell interference can be considered as log-normal ([11] and [20] page 1005). \( y_i \approx \ln(F_{i,m}^r) \) is thus Gaussian. We then have, in as much as 90% of the cases

\[
y_i < E[y_i] + 1.25 \sqrt{\text{Var}[y_i]}
\]

which gives

\[
I_{x,\kappa,m}^r < I_{x,\kappa,m}^{\text{sup}} = \frac{E[I_{x,\kappa,m}^r]}{\exp(1.25 \sqrt{\ln(E[I_{x,\kappa,m}^r]/E[I_{x,\kappa,m}^r]^2 + 1))}
\]

This limit depends mainly on the distance between the mobile and the base station, and not as much on the angle with a chosen reference axis [11]: \( I_{x,\kappa,m}^{r_{\text{sup}}} = I_{x}(r_{x,\kappa,m}) \).

Fig. 1. Cell Decomposition into rings

We now focus on the noise term. Although shadowing has a large effect in the other-cell interference term (see Eqn. 4), its direct effect in the noise term is not significant, as this noise term itself is small compared to the other terms. We can then consider \( f(r, \xi) \) with no shadowing in the noise term:

\[
f(r, \xi) \approx f(r) = \frac{(1 - \epsilon)W_B}{N} + \sigma^2 r^\alpha + \frac{I_x(r)}{N}
\]

Without loss of generality, we limit our study to two classes: \( V^x \) voice users requiring \( \gamma_v \), and \( D^x \) data users requiring \( \gamma_d \). CAC equation (5) becomes:

\[
\sum_{m=1}^{v} \gamma_v f(r_{x,v,m}) + \sum_{m=1}^{d} \gamma_d f(r_{x,d,m}) \leq W_B
\]

We define the effective bandwidth for a class-\( c \) mobile at distance \( r \) from its corresponding base station as: \( C^c(r) = \gamma_c f(r) \). As this effective bandwidth depends on the position of the mobile in the cell, location is a component of QoS. This leads to an infinite number of classes of users. To make this parameter finite and the analysis tractable, we divide the cell into \( n \) concentric circles of radii \( R_k, k = 1 \ldots n \), and define ring \( Z_k \) as the area between two adjacent circles of radii \( R_{k-1} \) and \( R_k \) (see figure 1-b). To obtain an equal partitioning of calls among rings, we take the areas of all zones equal to \( \pi(R_k)^2/n \).

For \( n \) large enough, function \( f(r) \) can be assumed constant over each ring \( Z_k \). The system is hence quantized. The immediate consequence of this quantization is that each class of users, voice or data, is divided into \( n \) subclasses depending on the user’s location in the cell. We then have \( 2n \) classes: \( n \) classes of voice calls, and \( n \) classes of data ones.

Define the effective bandwidth of user \( i \) of class \( c \) located in \( Z_k \) as \( C^c_k = \gamma_c f(R_k) \), where \( R_k \) is for instance equal to \( \sqrt{R_{k-1}R_k} \). CAC equation (8) gives then:

\[
\sum_{k=1}^{n} \left( C^v_k K^v_k + C^d_k K^d_k \right) \leq W_B
\]

where \( K^v_k \) voice calls and \( K^d_k \) data calls are present in ring \( Z_k \). This location-based CAC equation is an abstraction of the complexity of the WCDMA radio interface by a simple combination of a finite number of traffic classes. The state of cell \( x \) can then be defined by row vector \( \vec{s} \):

\[
\vec{s} := (K^v_1, ..., K^v_n, K^d_1, ..., K^d_n)
\]

and the space of admissible states \( A \) is defined as the subspace of \( \mathbb{R}^{2n} \) such that CAC condition (9) is verified.

C. Exact computation of the interference

The above study does not take into account the effect of the feedback between cells, nor the varying power emitted by the base stations. In this section, we calculate accurately the interference taking those two parameters into account.

According to Eqns. (3) and (4), the other-cell interference on a mobile station is a random variable depending on the random variables representing the interference on all the other mobile stations. It depends also on two independent random variables: the traffic density (number of users in each cell), and the spatial distribution of the mobiles (affecting their path losses). From Eqn. (3), the mean emitted power is:

\[
E[P^x] = \sum_{x\in A} \pi(\vec{s}) \sum_{k=1}^{n} (\pi^v_k E[Q^v_{x,k}] + \pi^d_k E[Q^d_{x,k}]) \sum_{m=1}^{v} \gamma_v (K^v_k) + \sum_{m=1}^{d} \gamma_d (K^d_k)
\]

where \( \pi(\vec{s}) \) is the probability of state vector \( \vec{s} \), calculated in the next section. Note that, contrary to the previous section, the mean emitted power depends on \( \pi(\vec{s}) \) which in turn depends on traffic load. Also, from Eqn. (4), the mean value of the other-cell interference in ring \( k \) of cell \( x \) is:

\[
E[I^x_k] = \sum_{y\neq x} E[P^y] E[\frac{Q^v_{y,k}}{Q^v_{y,k}}] + E[P^x] E[\frac{Q^v_{x,k}}{Q^v_{x,k}}]
\]

where \( Q^y_{x,k} \) is a random variable representing the path loss between base station \( y \) and a mobile located in ring \( Z_k \) of cell \( x \), and \( I^x_k \) is the other-cell interference in ring \( Z_k \) of cell \( x \). This expression takes into account the feedback between cells.

As of the variance \( \text{Var}[P^x] \), the high correlation between the interference in different zones makes its calculation very complicated. However, given that \( P^x \) is limited by \( W_B \) with mean value \( m^x = E[P^x] \), we can derive an upper bound on this variance based on the following relationship:

\[
\text{Var}[P^x] \leq E[(P^x)^2] - (m^x)^2 = \sum_{\vec{s} \in A} \pi(\vec{s})(P^x(\vec{s}) - m^x)^2
\]
Using the fact that $|P^x - m^x| < m^x$ if $m^x \geq \frac{W^B}{2}$ and is less than $W^B - m^x$ otherwise, the upper bound of the variance is:

$$\max(Var[P^x]) = \begin{cases} (W^B - m^x)^2 & m^x < \frac{W^B}{2} \\ (m^x)^2 & m^x \in \left[\frac{W^B}{2}, W^B\right) \\ (W^B - m^x)^2 & m^x \geq W^B \end{cases}$$

The variance of the other-cell interference is:

$$Var[I_k^x] = E[I_k^x]^2 - (E[I_k^x])^2$$

To evaluate the moments of the other-cell interference, $E[I_k^x]$ and $Var[I_k^x]$, we propose the following iterative algorithm:

**Algorithm I:**

1. Generate a series of system snapshots (a distribution of mobiles in different rings) and compute corresponding path losses. Compute then statistically the mean values:

$$E[I_k^x] = \sum_{y \neq x} E[I_y^x] \sum_{x,s,m} E[\sum_{y \neq x} \sum_{x,s,m} E[\sum_{y \neq x} \sum_{x,s,m} E[I_y^x]]]$$

2. Set the initial values of the effective bandwidth in CAC condition (9) by assuming that all surrounding base stations emit with their maximal power. The mean other-cell interference in the CAC condition is then

$$E[I_k^x] = W^B \sum_{y \neq x} E[I_y^x]$$

The variance is derived from Eqn. (12). The other-cell interference being log-normal, use in the CAC its upper bound as described in Eqn. (6).

3. Set the initial value of the other-cell interference for the power calculations in Eqn. (10) to 0.

4. Calculate the initial mean value for the powers $P^x$, using Eqn. (10).

5. Calculate the mean other-cell interference for all cells and rings using Eqn. (11).

6. Repeat steps 4) and 5) until the values converge. Calculate then the variance using Eqn. (12).

As these other-cell interference moments were obtained for a pessimistic CAC condition (step 2), we repeat steps 3-6 replacing the pessimistic value of $E[I_k^x]$ in Eqn. (13) by a new one between the initial value and the obtained one, until the other-cell interference and the effective bandwidth converge.

**Remark 1.** Algorithm I assumes that the cell is decomposed into rings as was the case for the simple model of the other-cell interference in the previous subsection. We will show by simulations in Section V that this decomposition is still valid.

**Remark 2.** When decomposing the cell into concentric rings, the last ring will cover parts of the adjacent cells to include calls that are in the soft handover region and are then connected to both cells at the same time. This will be also the case for irregular cell shapes, where we advocate the use of a circle large enough to cover the whole cell area and the soft handover zone. In this latter case, the overlapping zone will be larger than in the hexagonal case. The calls in the last ring will have a large interference factor which will be compensated by the macro-diversity gain. In fact, the signals originating from the two base stations will be combined in the mobile and the resulting SIR $\gamma_c$ will be the sum of the two achieved SIRs $\gamma_{c}^0$ in target cell and $\gamma_{c}^1$ in the adjacent cell [12]; $\gamma_c = \gamma_{c}^0 + \gamma_{c}^1$.

The effective bandwidth of these calls will then be equal to $C_n^c = \gamma_{c}^0 f(T_n^c)$.

### III. Priority CAC

#### A. CAC algorithm

Our first CAC algorithm aims to insure priorities between handoff and new calls, as well as between voice and data, while taking into account the effect of mobility. This is done by means of different acceptance ratios which we now define.

Calls arrive as new or handoff: voice handoff calls are given an absolute priority over all other calls, while data handoff calls are treated as new ones. Handoff voice calls are accepted if enough resources are available (i.e., condition (9) is verified taking into account that the handoff call is always situated in the last ring $Z_n$). For new calls of class $c$ arriving in ring $Z_k$, $k = 1...n$, they are accepted, if condition (9) is verified, with an acceptance ratio $a_k^c \leq 1$, $k = 1...n$. In doing so, a new call request may be blocked even though enough resources are available, in order to leave space to higher priority users. Handoff voice calls have an acceptance ratio of 1. Formally, the CAC algorithm is described in Figure 2.

**Remark 3.** Please note that, similar to [27], we consider that call handover blocking is not an important issue for data traffic, because data traffic can tolerate certain degree of delay/delay-jitter while voice cannot.

#### B. Markovian model

In what follows, we will focus on cell 0 and make use of the following assumptions:

A1) The arrival process of class-$c$ new calls is Poisson with rate $\lambda^c$ uniformly distributed over the cell surface.

A2) The mean arrival rate of class-$c$ handoff calls in ring $Z_n$ is equal to $\lambda^c_n$.

A3) The mean migrating rate of class-$c$ calls from ring $Z_k$ to ring $Z_j$, $j = k \pm 1$, is equal to $\lambda^c_{k,j}$.

A4) The service time of a class-$c$ call is exponentially distributed with mean $1/\mu^c$. 

![Fig. 2. CAC decision in the Priority CAC](image-url)
The arrival process of new class-\(c\) calls in ring \(Z_k\), \(k = 1, n\), is Poisson with rate
\[
\lambda_k^c = \lambda^c \frac{R_k^2 - R_{k-1}^2}{R^2} \quad (14)
\]
To simplify notations, let \(K_k^c\) denote the number of class-\(c\) calls within ring \(Z_k\), respectively, the system state is:
\[
\vec{s} := (K_1^c, ..., K_n^c, K_1^d, ..., K_d^d)
\]
**Definition 1.** \(A\) is the finite subset of \(\mathbb{N}^{2n}\) for which condition (9) holds. This means that a state vector \(\vec{s}\) is in \(A\) if and only if \(\sum_{k=1}^{n} (C_k^c K_k^c + C_k^d K_k^d) \leq W^B\).

Within the space of admissible states \(A\), transitions are caused by one of the following events:
1. Arrival of a new call in ring \(Z_k\), \(1 \leq k \leq n\). This transition is possible only if the call is accepted. Let us call the resulting state \(\vec{s}_k^c\).
2. Arrival of a handoff call in ring \(Z_n\), when there is room to accept it.
3. Termination of a class-\(c\) ongoing call in ring \(Z_k\). The next state is called \(\vec{s}_k^c\).
4. Migration of an ongoing class-\(c\) call from ring \(Z_k\) to ring \(Z_{k+1}\), \(1 \leq k < n\). Note that such a migrating call may be dropped if the next state \(\vec{s}_{k+1}\) is not within \(A\).
5. Migration of an ongoing call from \(Z_k\) to \(Z_{k-1}\), \(2 \leq k \leq n\). Note that a call migrating from ring \(Z_k\) to ring \(Z_{k-1}\) is never dropped. The next state is called \(\vec{s}_{k-1}\).
6. Departure of a class-\(c\) call from border ring \(Z_n\) to an adjacent cell.

**Definition 2.** \(A_k^c\) is the subspace of \(A\) where any other new call of class \(c\) in ring \(Z_k\) will be blocked due to lack of resources. In other terms, \(\vec{s} \in A_k^c\) if and only if \(\vec{s} \in A\) and \(\sum_{k=1}^{n} (C_k^c K_k^c + C_k^d K_k^d) + C_k^c > W^B\). \(A_k^c\) is the complementary subspace of \(A_k^c\) in \(A\).

**Definition 3.** \(A_{k+1}^c, k = 1, n - 1\), is the subspace of \(A\) where any migration of a call of class \(c\) from ring \(Z_k\) to \(Z_{k+1}\) will be blocked due to lack of resources. In other terms, \(\vec{s} \in A_{k+1}^c\) if and only if \(\vec{s} \in A\) and \(\sum_{k=1}^{n} (C_k^c K_k^c + C_k^d K_k^d) + C_{k+1}^c - C_k^c > W^B\).****

**C. Steady state probabilities**

If we consider that the dwell time of class-\(c\) mobiles in ring \(Z_k\) (i.e., the time they spend in ring \(Z_k\)) is exponentially distributed with mean \(1/\nu_k^c\), then the following theorem holds:

**Theorem 1.** The system described above is a Markov chain and the steady state probabilities are:
\[
\pi(\vec{s}) = \frac{1}{G} \prod_{k=1}^{n} \prod_{c=v}^{d} \left( \frac{\rho_k^c K_k^c}{K_k^c} \right), \quad \vec{s} \in A
\]
where \(\rho_k^c\) is the offered load of class-\(c\) calls in ring \(Z_k\):
\[
\rho_k^c = a_k^c \Lambda_k^c + a_k^c \nu_k^c I_{k=n} + \nu_{k+1,k}^c I_{k \neq n} + \nu_{k-1,k}^c I_{k \neq 1}
\]
(16)
where \(\eta_k^c = (\nu_k^c + \mu^c)\) and \(G\) is the normalizing constant:
\[
G = \sum_{\vec{s} \in A} \prod_{k=1}^{n} \prod_{c=v}^{d} \left( \frac{\rho_k^c K_k^c}{K_k^c} \right)
\]
(17)

**Proof:** Let \(B_k^c\) be the random variable indicating the activity time of a class-\(c\) call in ring \(Z_k\) and let \(X_k^c\) be the random variable denoting the call duration, and let \(Y_k^c\) be the random variable denoting the dwell time. \(X_k^c\) and \(Y_k^c\) are independent as the first depends on the amount of data to be sent and the latter on the mobility. We then have \(B_k^c = \min(X^c, Y_k^c)\) with cumulative distribution function (CDF):
\[
F_{B_k^c}(t) = P(X^c \leq t \text{ or } Y_k^c \leq t) = 1 - \exp(- (\mu^c + \nu_k^c) t)
\]
(18)
where \(F_{B_k^c}(t)\) is the CDF of the random variable \(V\) (pp. 141 in [25]). \(B_k^c\) is then exponential with mean \(1/\eta_k^c = 1/(\nu_k^c + \mu^c)\).

Our system corresponds to a BCMP (Baskett, Chandy, Muntz and Palacios) network [7] with a single node (corresponding to the cell), where class-\(c\) customers arrive from the outside to ring \(Z_k\) with rate \(\Lambda_k^c\) as new calls and \(\lambda_k^c\) as handoff ones. These rates must be multiplied by their corresponding acceptance ratios. These customers are served for a time equal to \(B_k^c\) and then either quit the queue (call termination or handoff), or reenter it after changing their class from \((c,k)\) to \((c,k+1)\), following certain routing probabilities [4] (migration from one ring to another). The total arrival rate of class-\(c\) calls in \(Z_k\) is then \(a_k^c \Lambda_k^c + a_k^c \nu_k^c I_{k=n} + \Lambda_{k+1,k}^c I_{k \neq n} + \Lambda_{k-1,k}^c I_{k \neq 1}\), and the load of class-\(c\) calls in \(Z_k\) is given by Eqn. (16).

According to the BCMP theorem for multiple classes of customers with possible class changes (see [7] pp. 146-150), all migrating rates are Poisson and the system is a Markov chain with the steady state probabilities having the product form given in Eqn. (15).

Note that this solution can be regarded as an approximation, as the migration rate between the different rings depends on the state of the system, through the number of calls. We will compare in the numerical applications this approximation with the exact solution defined in the next section and obtained by solving the balance equations.

**Remark 4.** It can be further shown that the formulas for the steady state probabilities extend also for the case where the service time and/or the dwell time for class-\(c\) calls in \(Z_k\) are not exponential, but have a general distribution. The system is no more a Markov chain, but Eqn. (15) still holds, as the steady state distribution in this case is insensitive to the call duration distribution and depends only on its mean [7]. This mean is derived via the resulting CDF of the activity time of class-\(c\) calls in \(Z_k\) (Eqn. (18)), which gives the density function and, by integration, the mean channel holding time \(1/\eta_k^c\).

**D. Handoff and migration rates**

So far, steady state probabilities (Eqn. (15)) have been given in terms of the load, itself given in terms of arrival rates, both migration and handoff \((\lambda_k^c\) and \(\eta_k^c\)) as well as new arrival rate \(\Lambda^c\). The latter is known. We now determine the former.

In equilibrium, the overall system may be assumed homogeneous and a cell statistically the same as any other one. We can thus assume that, for each class \(c\) of calls, the mean handoff arrival rate to the given cell is equal to the mean handoff departure rate from it [15], i.e., \(\lambda_k^c = \lambda_{n,n+1}^c\).
The migration rates from a ring \( Z_k \) to its adjacent rings depend jointly on the dwell times and the number of users in ring \( Z_k \). In fact, the number of active calls that move from a ring to its adjacent rings or to another cell increases when the ring is high loaded, or when mobiles have higher mobility. The migration rates for class-\( c \) calls are then:

\[
\lambda^c_{n,k+1} = \sum_{s \in A} \pi(s) K^c_{n,k} \nu^c_k \quad (19)
\]

\[
\lambda^c_{k,k+1} + \lambda^c_{k,k-1} = \sum_{s \in A} \pi(s) K^c_{k,k} \nu^c_k \quad k = 2, \ldots, n - 1 \quad (20)
\]

\[
\lambda^c_{k} + \lambda^c_{n,n-1} = \sum_{s \in A} \pi(s) K^c_{n,n} \nu^c_n \quad (21)
\]

On the other hand, if we assume that the trajectories followed by the different mobiles are randomly chosen, the migration rates from one ring are proportional to the perimeter of contact with its adjacent two rings:

\[
\lambda^c_{k,k+1} = \frac{R_k}{R_{k-1}} \lambda^c_{k-1,k-1}, \quad k = 2, \ldots, n \quad (22)
\]

One can see that these handoff / migration rates depend on the steady state probabilities, while the latter are themselves derived using the handoff / migration rates. To solve this problem, we use the following iterative algorithm that begins with an initial guess for handoff / migration rates.

**Algorithm II:**

1. Set initial values for the handoff / migration rates. To do so, we suppose that the blocking probabilities are negligible and that the blocking probabilities in Eqn. (25) are essentially due to the fact that the acceptance ratios \( a^c_k \) may be strictly less than 1:

\[
p^c_k \approx 1 - a^c_k
\]

If \( p^c_{k,j} \) is the probability that a call migrates from ring \( k \) to ring \( j = k \pm 1 \), we have:

\[
\lambda^c_{k,k+1}A^c_{k,k-1} \approx (p^c_{k,k+1} + p^c_{k,k-1})A^c_k c^c_k + \lambda^c_{k+1,k} + \lambda^c_{k-1,k}) \quad (23)
\]

The migration probability is the probability that the call quits ring \( Z_k \) before termination:

\[
p^c_{k,k+1} + p^c_{k,k-1} = P(Y^c_k < X^c_k) = \frac{\nu^c_k}{\nu^c_k + \mu^c} \quad (24)
\]

The initial migration/handoff rates are then obtained by solving the set of equations (23), using (22) and (24).

2. Calculate the other-cell interference and determine the admissible space \( \Lambda \) under the overall traffic load, using Algorithm I and the steady state probabilities (15).

3. Calculate the migration rates corresponding to the obtained state probabilities after convergence of Algorithm I using Eqns. (19)-(22).

4. Check the convergence of the migration rates using the relative error \( \epsilon = \max |1 - \frac{\lambda_{n,n}}{\lambda_{n,n}} |. \) If \( \epsilon > \epsilon \), where \( \epsilon > 0 \) is a predetermined constant, go to Step 2, otherwise, compute the performance measures.

**Remark 5.** The calculation of the migration / handoff rates in Step 1 is no more than the initial value for iteration. Even if the assumptions for blocking / dropping probabilities are not valid for heavy loaded systems, these values can be used. It is known that an iterative approach for solving the traffic equations converges for such a product form system ([7] pp. 330-335). Our numerical simulations (Section V-B) show that this algorithm converges after two to four iterations in each loop, which is slightly heavy. It is however feasible as it takes place off-line.

**E. Performance measures**

We now determine the performance measures relative to our system. We are interested in the dropping of ongoing calls, the blocking probabilities of new calls and the mean cell throughput.

1) Blocking probabilities: The first performance measure is the blocking probability. We obtain it by:

**Proposition 1.** The blocking probability \( p^p_k \) of a class-\( c \) call in ring \( Z_k \) is obtained by:

\[
p^p_k = 1 - a^c_k + a^c_k \sum_{s \in A^c_k} \pi(s) \quad (25)
\]

Specifically, the new call blocking probability is given by

\[
p^p_k = \frac{1}{\Lambda^c} \sum_{k=1}^n p^p_k \Lambda^c_k \quad (26)
\]

and the voice handoff call blocking probability is given by:

\[
p^h_k = \sum_{s \in A^c_k} \pi(s) \quad (27)
\]

**Proof:** A new connection of class \( c \) in ring \( Z_k \) is always blocked if the system is in a state \( \bar{s} \in A^c_k \). Otherwise, it is blocked with probability \( 1 - a^c_k \). In total, the blocking probability is

\[
p^p_k = (1 - a^c_k) \sum_{s \in A^c_k} \pi(s) + \sum_{s \in A^c_k} \pi(s) = 1 - a^c_k + a^c_k \sum_{s \in A^c_k} \pi(s)
\]

The overall blocking probability in the cell is directly derived by means of the relative arrival rates \( \Lambda^c_k / \Lambda^c \).

For voice handoff calls, they arrive only at the last ring \( Z_n \). They are then blocked when the system is in a state of the space \( A^c_n \) and the corresponding blocking probability is:

\[
p^h_k = \sum_{s \in A^c_n} \pi(s)
\]

Note that the blocking probability for handoff data calls is equal to that of new data ones at the cell border, i.e., \( p^h_k = p^p_k \).

2) Dropping probabilities: In the literature ([27], this term refers to the blocking of a handover call. We shall denote this particular event by inter-cell dropping. As we focus on intra-cell mobility, we should thus take into account the possibility of an intra-cell dropping event, i.e., a mobile station moving away from its base station experiences a higher interference figure and is thus dropped due to lack of resources. The overall dropping probability is hence the result of both intra and inter-cell dropping probabilities.
Lemma 1. The intra-cell dropping probability $d^e$ of a class-$c$ call due to mobility inside the cell is equal to:

$$d^e = \sum_{\vec{s} \in A} \sum_{k=1}^{n-1} K_k^e (v_k^e + \mu^e) \pi(\vec{s}) \sum_{n=1}^{A} \sum_{k=1}^{R_k} K_k^e \frac{R_k}{R_n + R_{n-1}} \gamma(\vec{s})$$

Proof: If the system is in state $\vec{s}$, a migrating class-$c$ call from ring $Z_k$ to ring $Z_{k+1}$ is dropped if the state $\vec{s}_{k+1}^R \notin A$. However, not all mobiles migrate: only those whose dwell time in the ring is less than their call duration time do migrate. The rate of leaving the ring $K_k^e (v_k^e + \mu^e)$ must then be multiplied by the probability that an ongoing class-$c$ call quits the ring:

$$P(Y_k^c < X^c) = \frac{v_k^e}{v_k^e + \mu^e}$$

Among these calls, only a fraction of $\frac{R_k}{R_n + R_{n-1}}$ try to migrate from ring $Z_k$ to ring $Z_{k+1}$. These calls are dropped if $\vec{s}_{k+1}^R \notin A$.

The intra-cell dropping probability $d^e$ due to mobility within the cell is the sum of dropping rates at each ring $Z_k$, divided by the rate of leaving the system, which gives the proof. ■

The overall dropping probability can now be determined by adding to the intra-cell dropping probability, the blocking rate of calls leaving the cell to adjacent ones, i.e., inter-cell dropping rate, divided by the rate of calls leaving the system. As the overall system is homogeneous in equilibrium, this blocking probability is equal to the handoff blocking probability calculated in Proposition 1. This leads to the following Proposition.

Proposition 2. The overall dropping probability $f^e$ of an ongoing class-$c$ call due to its mobility within the cell or between adjacent ones is equal to:

$$f^e = f^{e+1} = \sum_{\vec{s} \in A} \sum_{k=1}^{n-1} K_k^e (v_k^e + \mu^e) \pi(\vec{s}) \sum_{\vec{c} \in A} \sum_{k=1}^{R_k} K_k^e \frac{R_k}{R_n + R_{n-1}} \pi(\vec{s}) p_h^e$$

3) Throughput: Another important performance measure is the overall cell throughput. It is given by:

$$T = \sum_{\vec{s} \in A} \sum_{k=1}^{n} (K_k^e D^e + K_k^d D^d) \pi(\vec{s})$$

where $D^e$ is the throughput of a class-$c$ single user.

Remark 6. In heavy loaded areas, cells may be divided into three sectors by means of three directive transmitters in order to limit the interference. This introduces the notion of softer handover between sectors of the same cell. This softer handover is present in all rings and not only in ring $Z_n$. The softer handover blocking probability in ring $Z_k$ is then equal to $p_h^e$, replacing $a_h^e$ by 1 for voice calls. The same analysis can be applied to calculate the overall dropping probability. In particular, if $\lambda_{s,h}^e$ is the softer handover rate from ring $Z_k$ to the target sector to the adjacent sectors, we have:

$$\frac{\lambda_{s,h}^{e+1}}{2(R_k + R_{k+1})} = \frac{\lambda_{s,h}^{e+1}}{2\pi R_k} = \frac{\lambda_{s,h}^{e+1}}{2\pi R_{k+1}}$$

IV. SQUEEZING CAC

In UMTS, real-time flows keep a constant bit rate and hence a constant SIR figure over their duration. On the other hand, elastic traffic may be assigned variable bit rates (for example 32 Kbps, 64 Kbps, 128 Kbps, 144 Kbps, 384 Kbps, etc.), and hence variable SIRs. They nevertheless interfere with each other, influencing each other’s SIR.

In the previous section, we defined a preventive CAC scheme, where we block some low priority calls, although there are enough resources to accept them, in order to accept future high priority calls. In this section, we define a more realistic CAC algorithm that takes advantage of the elasticity of data traffic, and which we call Squeezing CAC. All arrivals are now accepted with their demanded SIR as long as there are resources ($a_k^e = 1, k = 1, ..., n, c = v, d$). If a call arrives and cannot be accommodated, we see if we can, by squeezing some or all ongoing data calls, accept it; it will be blocked otherwise. This squeezing can be done by slowing down the transmission of a subset of data calls, while voice users keep a fixed rate, as in [2][11]. In this case, squeezed data calls will experience lower transmission rates and longer call durations.

To choose the data calls to be squeezed, we begin by squeezing data calls with largest effective bandwidth requirements, and with larger distance to the base station, until we reach the first ring $Z_1$. Inside one ring, if $K_k^d$ data calls are in ring $Z_k$, among which $K_k^d$ are squeezed, and we need to squeeze more calls due to an arrival, we choose the calls to be squeezed randomly among the non squeezed existent calls. If now some squeezed data calls can be "upgraded" (set back to normal) due to a call departure, we choose randomly the data calls to be released among squeezed data calls with least effective bandwidth requirements.

To study the performance of the system, we need to define the elastic capacity in terms of the amount of data instead of the SIR. The relationship between the SIR $\gamma$ and the rate $D$ is given by [18]:

$$\gamma = D E_o / W$$

where $E_o/N_0$ is the despread bit energy per interference density and $W$ is the chip rate. Eqn. (31) indicates that when the rate of a call is divided by $c$, its SIR and its effective bandwidth are also divided by $c$. This squeezing factor $c$ is for instance equal to 2 if we define a data service with an initial bit rate of 128 Kbps, that can be squeezed to 64 Kbps. We can however consider other squeezing factors depending on the available transmission rates.

A. Exact solution

As it is possible to accept a call as long as enough resources are present, either initially or by squeezing some ongoing data calls by a factor $e$, the set of admissible states is the following:

Definition 4. $\vec{A}$ is the finite subset of $\mathbb{N}^{2n}$ for which $\sum_{k=1}^{n} (E_k^e K_k^e + E_k^d K_k^d) \leq W^B$, where $K_k^e$ is the number of voice calls and $K_k^d$ the number of data calls, be they squeezed or not, in ring $Z_k$.

Among all $K_k^d$ data calls, $K_k^d$ the number of squeezed data calls in ring $Z_k$, is determined by the squeezing policy
defined above. The corresponding system is non homogenous as the departure rate of elastic calls depends on the overall number of calls in the system. As a consequence, the product-form solution of the previous section no more holds although the system is still Markovian. To obtain the steady-state distribution of cell i, we must solve the equation \( \Pi \cdot Q = 0 \), knowing that \( \Pi T' = 1 \), where \( \Pi \) is the vector regrouping the steady-state probabilities \( \pi (T') \). \( Q \) is the generator matrix representing the transitions between neighboring states and \( T' \) is a vector of ones of proper dimension. Numerical resolution of this problem is possible, and once the vector \( \Pi \) is obtained, the performance measures can be obtained as in Section III-E.

To build the transition matrix \( Q \), we must consider all the possible transitions between the neighboring states, defined in Section III-B. We have:

\[
\begin{align*}
P(s \to s'_{k+1}) &= \lambda_k^c \rho^c_k \mu^c_k, \\
P(s \to s'_{k-1}) &= K_k^c \mu^c_k, \\
P(s \to s'_k) &= (K_k^c - K_k^d) \mu^d_k + K_k^d \mu^d_k/e. \\
P(s \to s'_k, -1) &= K_k^c \nu^c_k K_{k-1}^c + K_k^c K_{k-1}^c, \\
P(s \to s'_k, +1) &= K_k^c \nu^c_k K_{k+1}^c + K_k^c K_{k+1}^c. 
\end{align*}
\]

(32)

Note that the handoff rate must be taken into account for the last ring, and \( \bar{s'}_{k+1} = \bar{s'}_{k-1} = \bar{s} \), if migration is not possible due to dropping. The values \( P(s \to s') \) must be computed so that the sum of the terms in each line in the matrix is equal to zero. The iterations are needed in this solution only for calculating the handoff rates, as we take into account the migrations individually in each state.

B. Approximate product-form solution

In order to obtain a product-form solution, which in addition is less time-consuming than the exact one, we propose the following approximation:

**Proposition 3.** The steady state probabilities \( \pi (\bar{s}) \) can be approximated iteratively by

\[
\pi (\bar{s}) = \frac{1}{G} \prod_{k=1}^{n} \frac{(\rho^c_k K_k^c \mu^c_k) K_{k+1}^c}{K_k^c!), \quad \bar{s} \in \bar{A}
\]

(33)

\( G \) is the normalizing constant given by:

\[
G = \prod_{\bar{s} \in \bar{A}} \prod_{k=1}^{n} \frac{(\rho^c_k K_k^c \mu^c_k) K_{k+1}^c}{K_k^c!}
\]

(34)

\( \rho^c_k \) is the load of voice calls in ring \( Z_k \), defined in Theorem 4, and \( \bar{p}^d_k \) is the load of data calls in ring \( Z_k \) given by

\[
\bar{p}^d_k = \lambda^d_k \sum_{l=1}^{K_k} I_{k=l} + \lambda^d_{k+1,1} I_{k\neq n} + \lambda^d_{k-1,1} I_{k\neq 1} \frac{1}{\eta_k}
\]

(35)

\[
\eta_k = \frac{\hat{\alpha}_k}{\nu^c_k + \hat{\mu}_k} + \frac{1 - \hat{\alpha}_k}{\nu^e_k + \hat{\mu}_k}
\]

(36)

\[
\gamma_k = \frac{1}{\bar{\nu}_k} + \frac{1}{\bar{\nu}_k} \left[ \frac{\hat{\alpha}_k (\hat{\mu}_k + \nu^e_k)}{\mu^c_k (\mu^c_k + \nu^c_k)} + \frac{(1 - \hat{\alpha}_k)}{\mu^e_k (\mu^e_k + \nu^e_k)} \right]
\]

(37)

\[
\hat{\alpha}_k = \frac{(1 - \bar{p}^d_k)}{x_k^d y_k^d - x_k^d y_k^d} + \bar{p}^d_k (x_k^d y_k^d - x_k^d y_k^d)
\]

(38)

\( \mu^c_k \) and \( \mu^e_k \) are the eigenvalues of the matrix

\[
T_k = \begin{bmatrix}
-\mu^d_k & \bar{p}^d_k\mu^d_k \\
\bar{p}^d_k\mu^c_k & -\bar{p}^d_k\mu^e_k
\end{bmatrix}
\]

(39)

and \( (x_k^d, y_k^d) \) the corresponding eigenvectors.

\( \bar{p}^d_k \) is the probability that a data call enters ring \( Z_k \) in a squeezed state, \( \bar{p}^d_k \) is the probability that a non squeezed data call in ring \( Z_k \) be squeezed, and \( \bar{p}^d_k \) is the probability that a squeezed data call be upgraded, i.e. sees its transmission rate set back to normal:

\[
\bar{p}^d_k = \sum_{z_k} \pi (\bar{s})
\]

\[
\bar{p}^d_k = \sum_{z_{c,j}} \lambda^c \bar{p}^c_k \lambda^d \bar{p}^d_k
\]

\[
\bar{p}^d_k = \sum_{z_{c,j}} \lambda^c \bar{p}^c_k \lambda^d \bar{p}^d_k
\]

where \( \lambda^c, \lambda^d \) represent the arrival rate and the mean service time of class \( c \) calls in \( Z_j \), respectively.

The idea behind this approximation is that the service time of data calls is no more exponential, as they can be in different phases. They can enter the system in a squeezed state, denoted by state 2, with a certain probability \( \bar{p}^d_k \), or in a non squeezed state otherwise. In the squeezed state, a data call has an exponential service time of mean \( \frac{1}{\lambda^c} \), while its service discipline in the non squeezed state (state 1) is exponential with mean \( \frac{1}{\lambda^d} \). While being in the non squeezed state 1, this data call might finish its service and quit the system with probability \( 1 - \bar{p}^d_k \), or be squeezed and transit to state 1 with probability \( \bar{p}^d_k \). In state 1, a call finishes its service and quits the system with probability \( 1 - \bar{p}^d_k \). The service discipline of data calls can then be approximated by a Phase-type distribution PH-2 [24]. The values in Eqns. (36), (38) and 39) can then be obtained by a combination of the dwell time (exponential) and the service time (PH-2). On the other hand, migrating and handover calls have a certain age and their service times are the residual service times of PH-2 random variables, giving the value in Eqn. (37).

The remaining unknowns in the system are the squeezing probabilities \( \bar{p}^d_k \), calculated function of the steady state probabilities as described above. As these steady state probabilities depend in turn on the \( \bar{p}^d_k \)'s, this leads to an iterative calculation of the steady state probabilities. Note that this expression would be exact if the distribution of data service time is independent from the state of the system. However, because of the state-dependent nature of the system, this product-form corresponds to a heuristic that approaches the reality via the iterations.

V. NUMERICAL APPLICATIONS

The following assumptions and parameters shall be used in the numerical applications:

- The cell has a radius of 400 meters, and \( W^B = 10 \) watts. This corresponds to a standard base station having
A. Other-cell interference and cell decomposition

1) Validation of the decomposition into rings: Let us begin by validating our choice in dividing the cell into rings. To do so, we divide each ring into a number of sectors trying to exploit the symmetry of an homogeneous system. We apply Algorithm I to obtain the moments of the other-cell interference. The values in step 1 are obtained rapidly and used throughout the simulations. The algorithm consists of two loops: the main loop used to calculate the effective bandwidth in the CAC condition (steps 3-6), and the secondary loop used to calculate, given the effective bandwidth, the other-cell interference (steps 4-5). We observe that the secondary loop converges after 2 to 3 iterations, while the main one needs 3 to 5 iterations to converge, depending on the arrival traffic. We then consider that a decomposition of the cell into 4 rings represents a good solution and can be used as a base for calculating the other-cell interference over the cell surface. This simplifies further the issue of mobile location as rings are in this case large and an estimation error of the location is thus minimal.

2) Choice of the number of rings: Figure 4 shows the mean other-cell interference calculated for \( n = 3, 4, 5 \) and 6 rings. One can see that the curves are very close and form almost a single curve for \( n \geq 4 \) rings. This property was verified for \( n \) as large as 11 zones, and the relative error on the effective bandwidth obtained for \( n = 4 \) is less than 2%.

The standard deviation of the other-cell interference is drawn in Figure 5, and the same convergence can be observed. We then consider that a decomposition of the cell into 4 rings represents a good solution and can be used as a base for the CAC algorithm under general assumptions for the acceptance

<table>
<thead>
<tr>
<th>( n )</th>
<th>Ring 1</th>
<th>Ring 2</th>
<th>Ring 3</th>
<th>Ring 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.005</td>
<td>0.003</td>
<td>0.024</td>
<td>0.079</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>0.041</td>
<td>0.313</td>
<td>1.035</td>
</tr>
<tr>
<td>5</td>
<td>0.117</td>
<td>0.874</td>
<td>2.894</td>
<td>7.012</td>
</tr>
<tr>
<td>6</td>
<td>0.138</td>
<td>1.031</td>
<td>3.416</td>
<td>8.277</td>
</tr>
</tbody>
</table>

Table II: Variation of the other-cell interference with the traffic

B. Acceptance strategy for the Priority CAC

In Section III, we analyzed the performance of our Priority CAC algorithm under general assumptions for the acceptance
ratios. We now study it under four specific acceptance strategies which are:

1. No priorities between classes: a new call is accepted if there are enough resources, it is blocked otherwise, i.e., \( a_k^v = 1, \ k = 1...n, \ e = v, d \).
2. Voice calls are given absolute priority over data ones, i.e., \( a_k^v > a_k^d, \ \forall k \leq n, l \leq n \). We take for instance \( a_k^v = 1 \) and \( a_k^d = 0.95, \ k = 1...n \).
3. Data calls are given absolute priority over voice ones, i.e., \( a_k^d > a_k^v, \ \forall k \leq n, l \leq n \). We take for instance \( a_k^v = 0.95 \) and \( a_k^d = 1, \ k = 1...n \).
4. The priority of a class-c call is dependent on its effective bandwidth, i.e., \( a_k^v > a_k^d \) if \( E_k^v < E_k^d \). This leads to a situation where data calls near the base station have higher priority than distant voice calls. In our numerical applications for instance, the acceptance ratio near the base station is equal to 1 for both voice and data calls; it decreases linearly to reach 0.975 and 0.925 at the cell border for voice and data calls, respectively.

Algorithm II is used to calculate the performance measures. In addition to the two loops described in Algorithm I, a third loop is used to calculate the migration rates between the rings. We observe that this loop converges after 2 to 3 iterations.

Our aim is to minimize the dropping probability, at low voice blocking probability. We plot in Figures 6 and 7 the dropping probability and the voice blocking probability for the four strategies, respectively, function of the offered traffic (in Erlang).

We observe that giving higher priority to data calls (strategy 3) decreases the dropping probability. However, it is very harmful to voice calls as they will experience a high blocking rate. The lowest dropping probability is obtained when using strategy 4, where the blocking ratio is proportional to the effective bandwidth. This strategy achieves a low dropping probability at an acceptable blocking rate for voice calls. The only drawback is that this strategy results in a slightly lower throughput (see Figure 8), as calls with small rates have higher priority. This also may be unfair for users with large capacity demands. In the remainder of this work, we will adopt this strategy in our capacity calculations. However, the whole analysis holds for any chosen CAC strategy.

C. Investigating Erlang capacity

1) Priority CAC: The Erlang capacity is generally defined as the set of offered loads such that the corresponding blocking probability is smaller than a given \( \epsilon > 0 \) [2]. In this work, as we are also interested in the dropping probability, we will extend this definition to the following:

**Definition 5.** The Erlang capacity \( EC(\epsilon_1, \epsilon_2) \) is defined as the set of offered loads such that there exists a set of acceptance ratios \( (a_1^v, ..., a_n^v, a_1^d, ..., a_n^d) \) for which the dropping probability is less than a given \( \epsilon_1 > 0 \), and the blocking probability is less than \( \epsilon_2 \geq \epsilon_1 > 0 \).

In other words, we determine, if they exist, the acceptance ratios for each arrival rate that satisfy the constraints on blocking / dropping probabilities.

For illustration, we study the case where \( \epsilon_1 = 2\% \) and \( \epsilon_2 = 10\% \). We adopt acceptance strategy 4 where \( a_1^d = a_1^v = 1 \) and the acceptance ratios decrease linearly to reach \( a_{\text{min}}^v \) and \( a_{\text{min}}^d = (1 + a_{\text{min}}^d)/2 \), i.e. data calls near the cell border are accepted with an acceptance ratio of \( a_{\text{min}}^d \), while voice calls in the same conditions are accepted with a higher ratio equal to \( a_{\text{min}}^v = (1 + a_{\text{min}}^d)/2 \geq a_{\text{min}}^d \). We plot in Figures 9, 10 and 11 the dropping probabilities, the voice blocking probabilities and the throughput function of
One can see that the blocking probability increases when the acceptance ratio decreases, leading thus to a smaller number of accepted calls and a higher margin left for the mobility of existent users (a decrease of dropping probability).

On the other hand, in contrast with the throughput, the blocking probability increases rapidly with the arrival rate. This is because the other-cell interference, and thus the effective bandwidth, increases with the load. The space of admissible states is then reduced and more calls are blocked.

As of the Erlang capacity, we note that if no priorities were implemented, \( a_{\min}^v \), any arrival rate larger \( \Lambda = 0.047 \) calls per sec would not satisfy the performance measures. The dropping probability will be in this case larger than \( \epsilon_1. \) However, if we adopt a Priority-CAC strategy we can decrease dropping for larger arrival traffics. Consider for instance the case when \( \Lambda = 0.053 \) calls per sec, the constraint on the dropping probability is satisfied for \( a_{\min}^v \leq 0.87 \) (Figure 9), while the constraint for the blocking probability imposes that \( a_{\min}^v \geq 0.83 \) (Figure 10). The best choice that maximizes the throughput is then \( a_{\min}^v = 0.87 \) (Figure 11). Hence, the Priority CAC algorithm, by an appropriate choice of the acceptance ratios that exploits the trade-off between blocking and dropping probabilities, extends the Erlang capacity region to include additional arrival rates.

2) Squeezing CAC: To study the performance of squeezing CAC, we use Algorithm II, replacing the product form of the steady-state probabilities by a matrix inversion (Section IV-A). We solve the set of linear equations \( \Pi \cdot Q = 0 \), knowing that \( \Pi \cdot \overline{e} = 1 \) using the matlab function ‘linsolve’. For the seek of comparison, we also solve the system using the product form approximation defined in Section IV-B.

We plot in Figure 12 the overall dropping probability for different arrival rates as a function of the squeezing factor \( c \). One can see that this combined intra-cell and inter-cell dropping probability decreases when the squeezing factor increases. A similar behavior is observed for the voice blocking probability in Figure 13. This is very useful for the Erlang capacity region, as both dropping and blocking decrease.

As of the throughput, drawn in Figure 14, it increases when data calls are squeezed. This is explained by the intelligent use of resources: a data call is squeezed only if we need to do so in order to accommodate a new arrival. Resource utilization, and consequently throughput, is then higher.
D. Comparison

We now compare our two CAC algorithms to the basic CAC algorithm, where no priorities are implemented \((a_k^c = 1, k = 1, \ldots, n, c = v, d)\) and no squeezing is performed \((e = 1)\).

The Priority CAC makes use of preventive blocking to insure priorities between flows. In doing so, the dropping probability decreases, at the price of a larger blocking probability. This leads to an extension of the Erlang capacity region by a compromise between blocking and dropping rates.

Squeezing CAC is a state dependent CAC, where squeezing is only performed if there is a need to accept a new arrival; there is no preventive blocking as in Priority CAC. Squeezing CAC can also be used to ameliorate Erlang capacity, as it decreases both dropping and blocking. It also increases the throughput. The only drawback is that the mean duration of data calls, and thus the file download time increases. For an arrival traffic of 0.053 for instance, we observe an increasing of the mean download time of 5% for a squeezing factor of 1.35 and of 21% for a squeezing factor of 2.

In summary, Squeezing CAC with small squeezing factor is preferable to increase cell throughput and decrease blocking and dropping, without increasing excessively download times.

1) Approximation error: We plot in Figure 15 the blocking and dropping rates obtained for an arrival traffic equal to 0.033 for the squeezing CAC. The Figure compares the results obtained with the exact (Section IV-A) and approximate (Section IV-B) solutions. One can see that the approximation is good and gives small errors of the order of 6% for dropping and 16% for blocking. Note that this approximation error exists even for a squeezing factor equal to 1 (no squeezing), as the analysis in Theorem 1 approximates the state-dependent migration-rates by state-independent mean values.

E. Localization issues

When taking our CAC decision, we need to localize the mobile users. Several localization algorithms were developed in the literature, and they are used by many vital services in mobile networks, such as emergency call assistance, navigation, location-based value-added services, etc. The most accurate ones are based on Global Positioning System (GPS) [10], where GPS receivers are added to the mobile terminals, or on Satellite-assisted Positioning based on differential GPS [19]. As GPS receivers are not yet implemented in all terminals, many network-based techniques have then been developed to
track mobile’s position without using GPS. Some of these methods are based on the Angle of Arrival (AOA) method [8], where the mobile’s signal is received at various antenna sites, each of them equipped with an adaptive antenna array to detect the compass direction from which the caller’s signal is arriving. Other network-based algorithms use pattern recognition based on hidden Markov chains in order to obtain good estimates from the Time of Arrival (TOA) measurements [22].

The most accurate network-based methods are those based on the Time Difference of Arrival (TDOA) method [6][19][23], where location is derived from the intersection of two lines of positions, each of them in turn derived from the measured time difference of the arrivals of the signals originating from two different base stations. Different techniques can be applied to extract information from TDOA measurements, but it is shown in [23] that Kalman filtering allows a better UMTS mobile tracking than a static location estimator based on Cramer Rao Bound (CRB) [6]. The positioning error for Kalman filtering has been shown to be on the order of 7 meters [23]. We hence suggest the use of TDOA-based Kalman filtering for user localization. This should be accurate enough in our case since we advocate the use of limited number of rings only, say 4 rings, as shown in Section V-A.2.

To study the impact of localization error on the performance of our CAC, we suppose that the localization algorithm induces a random estimation error. A certain number of arrivals in a given ring may thus be wrongly considered as belonging to an adjacent ring. Let \( \alpha_{k,j}^c \), \( j = k \pm 1 \), be the rate of class-\( c \) calls that arrive in ring \( Z_k \) and that are wrongly considered as belonging to ring \( Z_j \). Some of them may hence be subject to a wrong CAC decision. \( \alpha_{k,j}^c \) is proportional to the arrival traffic \( \Lambda \) and the distribution of the localization error.

For the priority-CAC for instance, an acceptance ratio of \( \alpha_{k,j}^c \) is applied to the \( \alpha_{k,j}^c \) calls, instead of the desired acceptance ratio \( \alpha_{k,j} \). Even if we suppose that \( \alpha_{k,j}^c \simeq \alpha_{k,j} \), we will have the same overall preventive blocking, but the occupied capacity increases because of the difference between the effective bandwidths in the two adjacent zones, i.e., wrongly accepted calls have higher effective bandwidth then the wrongly rejected ones. We hence operate in a higher load condition than initially assessed. This results in a higher dropping probability and a higher blocking rate due to lack of resources. To quantify this phenomena, let \( \bar{\Lambda}_k \) be the effective class-\( c \) arrival rate in \( Z_k \), considering that some \( \alpha_{k,j} \) CAC decisions were taken due to the localization error. If \( \bar{p}_k^c \) denotes the new class-\( c \) blocking probability in \( Z_k \), the effective arrival rate is approximated by:

\[
\bar{\Lambda}_1^c \simeq \Lambda_1^c - (p_1^c - \bar{p}_1^c)\alpha_{12}^c \\
\bar{\Lambda}_k^c \simeq \Lambda_k^c + (p_k^c - \bar{p}_k^c)\alpha_{k,k-1}^c - (p_{k+1}^c - \bar{p}_{k+1}^c)\alpha_{k+1,k}^c, \quad k = 2, n-1 \\
\bar{\Lambda}_n^c \simeq \Lambda_n^c + (p_n^c - \bar{p}_n^c)\alpha_{n,n-1}^c - (1 - p_n^c)\alpha_{n,n+1}^c
\]

where \( \alpha_{n,n+1}^c \) is the rate of calls arriving in the last ring \( Z_n \) but wrongly considered as belonging to an adjacent cell due the localization error. Note that some calls arriving to adjacent cells may also be wrongly considered as arriving to the target cell, thus creating a new class of calls with higher effective bandwidth \( C_{n+1}^c \). As we consider a homogeneous system, their arrival rate will be equal to \( (1 - p_n^c)\alpha_{n+1,n}^c \).

We compare in Figure 16 the dropping and blocking probabilities for a perfect location estimation and for a location estimation with a uniformly distributed error ranging between 0 and 50 meters. We can see that, in accordance with our analysis, both the blocking and dropping probabilities increase due to the localization error. They however follow the same trend, i.e. when the acceptance ratio increases, the blocking probability decreases, while the dropping rate increases. This allows the same extension of the Erlang capacity, as in Section V-C. Note that the two curves, i.e. obtained with error-free and errored location estimation, become closer as the acceptance ratios increase. This occurs when the blocking becomes less preventive and the dependence between the CAC decision and the location through the acceptance ratios decreases (the blocking rate increasing ranges from 4% to 12%).

![Fig. 16. Effect of the localization error on the blocking and dropping probabilities in the Priority CAC](image)

VI. CONCLUSION

In this paper, we developed two mobility-aware admission control schemes for the downlink of WCDMA mobile networks. We first calculated the other-cell interference taking into account the feedback between cells, and obtained effective bandwidth expressions based on both the multimedia class and the location of the user in the cell. This location is subject to change, hence accounting for the user’s mobility pattern, and is modeled by dividing the cell into concentric rings.

Based on this model, we developed two CAC algorithms. In the first priority-based one, we attributed to each class of users an acceptance ratio to handle priorities between flows, and proved that the underlying system can be modeled as a Markov chain. We then obtained the steady state probabilities and gave an iterative algorithm to determine them explicitly. The second CAC algorithm, called squeezing CAC, is state dependent: all arriving calls are accepted until the system becomes saturated. Saturation is reached when the system is full and either a new call arrives or an already connected mobile moves away from the base station. In this case, some existing data calls are squeezed so as to alleviate saturation.

As of the performance measures, we obtained expressions for the blocking probabilities and the dropping probabilities...
of ongoing calls, both intra-cell due to their mobility within the cell and inter-cell due to handoff. We also determined the overall cell throughput.

Our simulations show that the cell decomposition into rings is valid and that four rings are sufficient. We found out that a higher traffic load yields a larger other-cell interference and thus reduces the space of admissible states. We then studied the performance of our CAC algorithms, along with the Erlang capacity, and showed how to increase the Erlang capacity region by controlling blocking and dropping rates.

References


