

Weight allocation in distributed admission control for wireless networks

Youssef Iraqi*, Raouf Boutaba

School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ont., Canada N2L 3G1

Received 11 November 2003; revised 29 September 2004; accepted 30 September 2004
Available online 10 November 2004

Abstract

In this paper, we introduce a weight allocation strategy used to combine the received information from a set of base stations involved in a distributed admission control process. A method to compute the weights in a one and two-dimensional networks is proposed. We also propose a Distributed Call Admission Control (DCAC) framework designed for wireless mobile multimedia networks. We evaluate the performance of the DCAC scheme in terms of call-dropping probability, call blocking probability and average bandwidth utilization. We further introduce a combined performance metric to facilitate performance comparison between CAC schemes. Simulations demonstrate that the weight allocation strategy improves the performance. We also investigate the impact of the number of involved cells in the admission control process on the overall performance.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Weight allocation; Distributed admission control; Wireless networks

1. Introduction

1.1. Background

As the mobile network is often simply an extension of the fixed network infrastructure from the user's perspective, mobile wireless users will demand the same level of service from each. Such demand will continue to increase with the growth of multimedia computing and collaborative networking applications. This raises new challenges to call (session) admission control (CAC) algorithms.

Furthermore, the (wireless) bandwidth allocated to a user will not be fixed for the lifetime of the connection as in traditional cellular networks, but rather, the base station will allocate bandwidth dynamically to users. Many evolving standards like UMTS [1] have proposed solutions to support such capability.

1.2. Related works

Call admission control schemes can be divided into two categories, local and collaborative schemes [2]. Local schemes use local information alone (e.g. local cell load) when taking the admission decision. Examples of these schemes are [2–5,6]. Collaborative schemes involve more than one cell in the admission process. The cells exchange information about the ongoing sessions and about their capabilities to support these sessions. Examples of these schemes are [7–13].

The fundamental idea behind all collaborative admission control schemes is to consider not only local information but also information from other cells in the network. The local cell, where the new call has been requested, communicates with a set of cells that will participate in the admission process. This set of cells is usually referred to as a cluster. In general, the schemes differ from each other according to how the cluster is constructed, the type of information exchanged and how this information is used.

In [14] for example, the cluster is defined as the set of direct neighbors. The main idea is to make the admission

* Corresponding author. Tel.: +1 888 4567x2747; fax: +1 885 1208.

E-mail addresses: iraqi@bcr.uwaterloo.ca (Y. Iraqi), rboutaba@bcr.uwaterloo.ca (R. Boutaba).

decision while taking into consideration the number of calls in adjacent cells, in addition to the number of calls in the local cell. In [15,16], authors have also defined the cluster to be the set of direct neighbors. In another work, Levine et al. [17] extended the basic distributed scheme by embedding mobility modelling and dynamic cluster. This scheme is based on the shadow cluster concept [9]. The shadow cluster is constructed using information about user mobility parameters. Aljadhai et al. [12] developed their admission control based on the most likely cluster concept. The concept of directional probabilities is introduced to build the most likely cluster. The probabilities are based on user mobility information similarly to [17]. In [4,18], the cluster is defined as being the cells up to three hops away from the central cell handling the mobile.

Commonly in distributed admission control, the local cell sends a request to other cells in the cluster and then, after receiving the requested information, makes its final decision. However, the way these responses are combined did not receive much attention in previous research. In [14], a new call request is accepted if the overload probabilities of the original cell and ALL the cells in the cluster are below a specified threshold. In [13,16], admission is granted to a new request if ALL the cells in the cluster can afford to reserve a particular amount of bandwidth.

In [17], the cell receiving the admission request receives a set of values from neighboring cells (called availability estimates) that indicate their level of congestion. The cell then takes an average value (called survivability estimates) and accepts users with a survivability estimate higher than a particular threshold. The cell uses active mobile probabilities to weight the responses from neighboring cells. These probabilities are computed based on the user movement and represent the probability of visit to a particular cell.

In [19], the same weight is given to all neighboring cells. Newly arriving users are admitted into the system provided that the predicted single probability of dropping any user in the home and neighboring cells is below a pre-specified threshold. The scheme proposed by Aljadhai in [12], provides two kinds of predictive services: integral guaranteed service and fractional guaranteed service. In the integral guaranteed service ALL cells in the cluster must support the requested bandwidth for the lifetime of the call. In the fractional guaranteed service, a call is accepted if at least $\gamma\%$ of the cells in the cluster can support at least $\lambda\%$ of the requested bandwidth. Here as in [17] the cells are weighted according to the probability of visit.

1.3. Motivation and contribution

In all studied schemes, weights are not assigned judiciously to the responses of the cells involved in the CAC scheme. We believe that, such weight allocation is crucial to the performance of any scheme. To understand

why, let us take the following example: assume that a mobile terminal going in a linear direction requests admission to cell 1. Assume that the cluster is composed of three cells in the direction of the mobile, say cells 1, 2 and 3. Now assume that cell 2 is congested and cannot accept the user (not now and not at any future time). The information that cell 1 will receive from cell 3, is irrelevant to the admission of the user even if cell 3 has enough bandwidth to accept the user. Because two out of three cells can accept the user is not sufficient for this user to be admitted into the network. Also, because the user is moving in a given direction, there is no use in reserving bandwidth in a cell that is in the opposite direction. It is hence crucial to judiciously assign weights to the various cells in a cluster in order to make the right admission decision.

In this paper, we introduce a novel method for combining the responses of the cells involved in the admission process by associating to each cell a carefully chosen weight. The weight assignment method is then incorporated into a Distributed Call Admission Control (DCAC) scheme we propose. The DCAC performance is evaluated and compared with the Guard Channel [7] and the Shadow Cluster [17] schemes. We demonstrate that with the weight allocation strategy the DCAC scheme has better performance in terms of call-dropping probability (CDP), call-blocking probability (CBP) and average bandwidth utilization (ABU).

The paper is organized as follows. In Section 2, we describe the model of the system considered in this paper. Section 3 defines the concept of dynamic mobile probabilities. Section 4 introduces the weight allocation strategy. Section 5 presents the distributed call-admission-control scheme involving a cluster of neighboring cells. In Section 6 we present the call-admission control performed locally by the cells in our system. Section 7 gives the detailed steps of the distributed admission control algorithm, and Section 8 introduces a new combined QoS metric for CAC schemes. Sections 9 and 10 discuss the simulations conducted and present a detailed analysis of the obtained results. Finally, Section 11 concludes the paper.

2. System model

We consider a wireless network with a cellular infrastructure that can support mobile terminals running applications that demand a wide range of resources. Users can roam the network freely and experience a large number of handoffs during a typical connection. We assume that users have dynamic bandwidth requirements. The wireless network must provide the requested level of service even if the user moves to an adjacent cell. A handoff could fail due to insufficient bandwidth in the new cell (or in a neighboring cell if a mechanism like the directed retry [20] is used), and in such case, the connection is dropped.

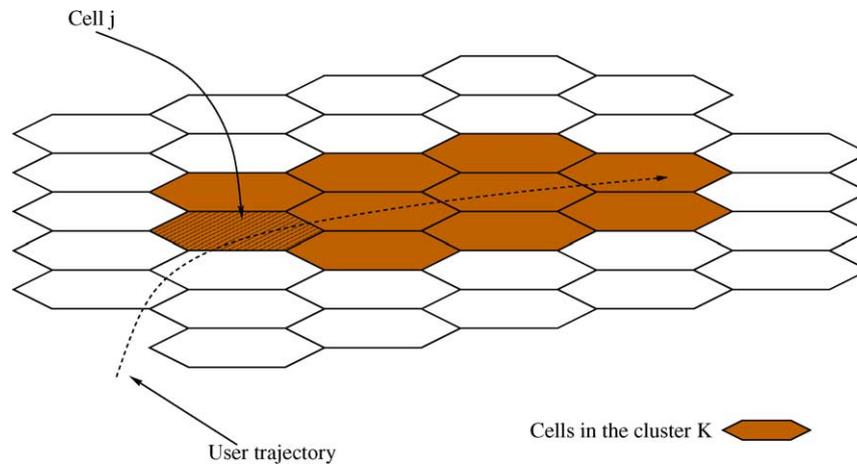


Fig. 1. Cell j and the cluster for a user.

To reduce the call-dropping probability, we make neighboring cells participate in the admission decision of a new user. Each cell will give its local decision and then the cell where the request was issued will decide if the new request is accepted or not. By doing so, the admitted connection will more likely survive handoffs.

As any distributed scheme, we use the notion of a cluster or group of cells (see Fig. 1). Each user in the network with an active connection has a cluster associated to it.¹ The cells in the cluster are chosen by the cell where the user resides. The shape and the number of cells of a user's cluster depend on factors such as the user's current call-holding time, QoS requirements, terminal trajectory and velocity.

3. Dynamic mobile probabilities

We consider a wireless network where time is divided into equal intervals at t_0, t_1, \dots, t_m where $\forall i \geq 0 \ t_{i+1} - t_i = \tau$. Let j denote a base station in the network,² and x a mobile terminal with an active wireless connection. Let $K(x)$ denote the set of cells that form the cluster for user x . We write $[P_{x,j,k}(t_0), P_{x,j,k}(t_1), \dots, P_{x,j,k}(t_{m_x})]$ for the probability that mobile terminal x , currently in cell j , will be active in cell k , and therefore under the control of base station k , at times $t_0, t_1, t_2, \dots, t_{m_x}$. These probabilities are named differently by different researchers, but basically they represent the projected probabilities that mobile terminal x will remain active in the future and at a particular location. It is referred to as the Dynamic Mobile Probability (DMP) in the following. The parameter m_x represents how far in the future the predicted probabilities are computed. It is not fixed for all users and can depend of the user's QoS or the actual elapsed time of the connection.

¹ In this paper the term 'user' and 'connection' are used interchangeably.

² We assume a one-to-one relationship between a base station and a network cell.

DMPs may be functions of various parameters such as the handoff probability, the distribution of call duration for a mobile terminal x when using a given service class, the cell size, the user mobility profile, etc. The more information we have, the more accurate the probabilities, but the more complex is their computation.

For each user x in the network, the cell responsible for this user determines the size of the cluster $K(x)$. The cells in $K(x)$ are those that will be involved in the CAC process. The cell responsible for user x sends the DMPs to all members in $K(x)$ specifying whether the user is a new one (in which case the cell is waiting for responses from the members of $K(x)$).

DMPs range from simple probabilities to complex ones. A method for computing dynamic mobile probabilities taking into consideration mobile terminal direction, velocity and statistical mobility data, is presented in [9]. Other schemes to compute these probabilities are presented in [10,13]. To compute these probabilities, one can also use mobile path/direction information readily available from certain applications, such as the route guidance system of the Intelligent Transportation Systems with the Global Positioning System (GPS) [21]. In this paper, we assume that these probabilities are computed as in [17], however, the proposed weights allocation strategy and admission control can use other methods to compute these probabilities as more precise and accurate methods become available.

4. Weights allocation strategy

Let us assume for now that each cell k in the cluster $K(x)$ sends a response $R_k(x)$ to tell the local cell j about its ability to support user x , and assume that $R_k(x)$ is a real number between -1 (i.e. cannot accept user x), and $+1$ (i.e. can accept user x). Here, the admission decision takes into account the responses from all the cells in the user's

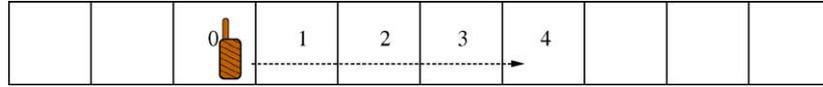


Fig. 2. An example of a highway covered by 10 cells.

cluster $K(x)$. The cell has to combine the responses $R_k(x)$ and takes the final decision regarding the admission request. The cell has to decide the weight of each cell k in the user's cluster $K(x)$. This will define the contribution of each cell to the final decision.

We have identified two factors for determining the weight of each cell in $K(x)$: the *temporal relevance* and the *spatial relevance*.

4.1. Temporal relevance

If a cell $k1$ in the user's cluster supports the user more than another cell $k2$, cell $k1$ should have a higher impact on the admission of user x than cell $k2$. In general, the longer a cell is involved in supporting the user, the higher its impact. The temporal relevance $T_k(x)$ represents this impact. We propose the following formula for computing the temporal relevance $T_k(x)$ of cell k

$$T_k(x) = \frac{\sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k}(t)}{\sum_{k' \in K(x)} \sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k'}(t)} \quad (1)$$

This is the ratio of the sum over time of the DMPs when the mobile is in the considered cell k , over the sum of all the DMPs for all cells in the cluster. This parameter gives an indication of the percentage of time the user may spend in the considered cell k relative to the time the user is spending in the cluster. Eq. (1) can be computed by the local cell j based only on the dynamic mobile probabilities.

4.2. Spatial relevance

To explain the idea of spatial relevance, we use the following example. Consider a linear highway covered by 10 square cells as in Fig. 2. Assume that a new user, following the trajectory shown requests admission in cell number 0 and that the CAC process involves five cells. Responses from cells numbered 1–4 are relevant only if cell 0 can accommodate the user. Similarly, responses from cells 2–4 are relevant only if cell 1 can accommodate the user when it hands off from cell 0. This is because; a response from a cell is irrelevant if the user cannot be supported on the path to that cell. We note $S_k(x)$ the spatial relevance of cell k for user x .

$S_k(x)$ depends only on the topology of the cellular network and the responses from other cells in the cluster. For the sake of clarity, we will consider in the following a one-dimensional network first.

4.2.1. One-dimensional case

For the linear highway example of Fig. 2, we propose the following formula to compute the spatial relevance

$$S_0(x) = 1 \text{ and } S_k(x) = \prod_{l=1}^k f(R_{l-1}(x)) \quad (2)$$

where $f(R) = \frac{(1+R)}{2}$.

This formula is chosen so that if one of the cells l before cell k has a negative response (i.e. $R_l(x) = -1$), the spatial relevance of cell k is 0; and if all of the cells l before cell k have a positive response (i.e. $R_l(x) = 1$), the spatial relevance of cell k is 1. Note that for each $k \in K(x)$ we have $0 \leq S_k(x) \leq 1$. Note also that in Eq. (2), cell j (the cell receiving the admission request) has the index 0 and that the other cells are indexed in an increasing order according to the user direction as in Fig. 2. We have chosen $f(R)$ to be $\frac{(1+R)}{2}$, however, the only requirement is that it should be an increasing function with $f(-1) = 0$ and $f(1) = 1$. f will influence the effect that the responses from previous cells will have on the spatial relevance of the cell.

4.2.2. Two-dimensional case

We consider a 2D network as shown in Fig. 3, where the number inside each cell denotes the cell number. We will

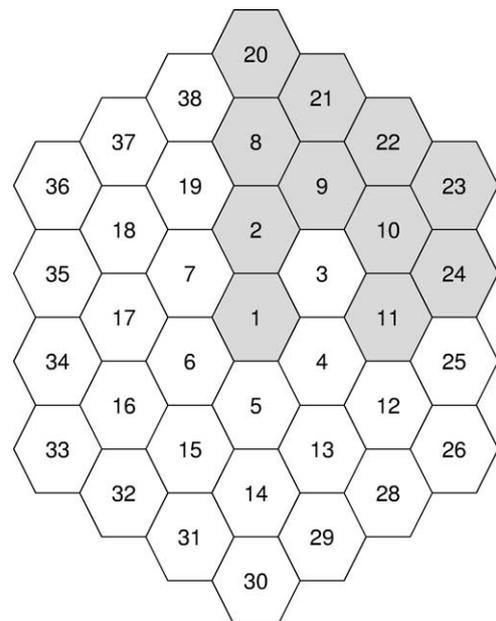


Fig. 3. A two-dimensional network.

Table 1
Possible paths from cell 1

Destination	2	8	9	10	11	20
Path 1	2	2,8	3,9	3,10	4,11	2,8,20
Path 2	–	–	2,9	–	3,11	–
Path 3	–	–	–	–	–	–

assume that the user is in cell number 1, so that all spatial relevance degrees will be computed relative to the position of this cell.

To compute the spatial relevance for a particular cell k , we will need to know what are the possible paths that the user can take to reach cell k from cell 1. We will assume that only the shortest paths from cell 1 to cell k are considered. For symmetrical reasons, only paths from cell 1 to any cell in the gray area in Fig. 3 will be presented. Paths to other areas in the network can be derived by symmetry. Tables 1 and 2 show for each of the gray cells the possible paths. Of course, there is no shortest path from cell 1 to cell 1, and the spatial relevance of cell 1 is 1. Note that possible paths can also be derived for cells that are more than two cells away from cell 1.

To compute the spatial relevance degrees, let us take the following example: Assume that there is a path π_1 between cell 1 and cell k such as $\pi_1=(1, a, k)$ (meaning that the user has to go through cell a to reach cell k). As in the 1D case, $S_k(x)=S_a(x)f(R_a(x))$ with f defined as before. Also, $S_a(x)=S_1(x)f(R_1(x))$ since $S_1(x)=1$. Hence we have, $S_k(x)=f(R_1(x))f(R_a(x))$. We can define then, for each path $\pi=(1, a_1, a_2, a_3, \dots, a_n)$ from cell 1 to cell a_n , the spatial relevance relative to path π as follows

$$S_{a_n, \pi}(x) = f(R_1(x)) \prod_{l=1}^{n-1} f(R_{a_l}(x)) \quad (3)$$

We can then define the spatial relevance for a cell k as follows

$$S_k(x) = \frac{\sum_{\pi \in \mathcal{Q}_{1,k}} S_{k, \pi}(x)}{\mathcal{Q}_{1,k}} \quad (4)$$

where $\mathcal{Q}_{1,k}$ is the set of possible shortest paths from cell 1 to cell k , and $\mathcal{Q}_{1,k}$ is the number of paths in the set.

Note that Eq. (4), if applied to a 1D network, will lead to Eq. (2).

Table 2
Possible paths from cell 1

Destination	21	22	23	24
Path 1	3,9,21	3,10,22	3,10,23	3,11,24
Path 2	2,9,21	3,9,22	–	3,10,24
Path 3	2,8,21	2,9,22	–	–

Now that we have defined the weight allocation strategy, we will present, in the following, a distributed admission control algorithm that utilizes this weight allocation strategy. Note that the proposed allocation strategy can be used by other distributed admission control schemes.

5. The distributed admission control process

In this distributed admission control algorithm, the cell receiving the admission request computes the sum of the product of $R_k(x)$, $T_k(x)$ and $S_k(x)$ over k . The final decision of the call admission process for user x is based on

$$D(x) = \frac{\sum_{k \in K(x)} R_k(x) T_k(x) S_k(x)}{\sum_{k' \in K(x)} T_{k'}(x) S_{k'}(x)} \quad (5)$$

Note that $-1 \leq D(x) \leq 1$ and that $\sum_{k' \in K(x)} T_{k'}(x) S_{k'}(x)$ is never 0, since the spatial relevance, $S_j(x)$, of cell j is always equal to 1, its temporal relevance $T_j(x)$ is strictly positive, and all other $S_{k'}(x)$ and $T_{k'}(x)$ are positive or 0.

If $D(x)$ is above a certain threshold, called acceptance threshold (T_{acc}), user x is accepted, otherwise, the user is rejected. The higher $D(x)$, the more likely the user connection will survive in the event of a handoff.

6. Local admission control process

We show here how $R_k(x)$ are computed. We assume that user's traffic can be voice, data or video. Voice users are usually characterized by a fixed bandwidth demand. Data and video users have a dynamic bandwidth requirement due to the burstiness of data and video traffic. Without loss of generality, we can assume that a user x is characterized by a bandwidth demand distribution $f_x(E_x(c), \sigma_c)$, where $E_x(c)$ and σ_c are the mean and the standard deviation of the distribution f_x , respectively, and c is the type of traffic for user x . Note that $E_x(c)$ depends on the traffic type c (voice, data or video). More service classes can be defined if required.

6.1. Computing elementary responses

At each time t_0 , each cell in a cluster $K(x)$ involved in our CAC process for user x makes a local CAC decision for different times in the future (t_0, t_1, \dots, t_{m_x}). Based on these CAC decisions, which we call 'elementary responses,' the cell makes a final decision that represents its local response to the admission of user x to the network. Elementary responses are time-dependent. The computation of these responses varies according to the user location and type.

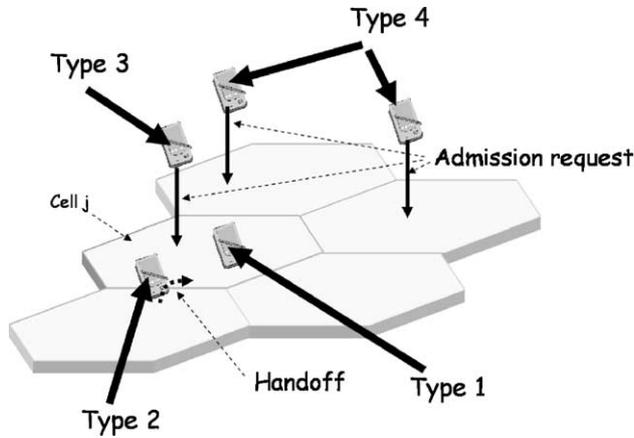


Fig. 4. User types.

6.1.1. User types

A cell may be involved in processing different types of users. Possible user types at time t_0 are (see Fig. 4):³

- (1) Old users local to the cell,
- (2) Old users coming from another cell (executing a handoff),
- (3) New users (at time t_0) within the cell, or
- (4) New users (at time t_0) in other cells.

New users are defined as all users seeking admission at time t_0 . Users of type 1 have the highest priority. Priority among other users is subject to some ordering policy. The network tries to support old users if possible and uses the DMPs to check if a cell can accommodate a new user who will possibly come to the cell in the future.

6.1.2. Local CAC at time t_0 for time t_0

The cell can apply any local call admission algorithm to compute the elementary responses. In this work we assume that the cells use the equivalent-bandwidth approach to compute these responses.

6.1.3. Local CAC at time t_0 for time t_l ($t_l > t_0$)

Each base station computes the equivalent bandwidth at different times in the future according to the DMPs of future users.

Assume user x , in cell j at time t_0 , has a probability $P_{x,j,k}(t_l)$ of being active in cell k at time t_l and has a bandwidth demand distribution function f_X .

Define $Y = \Phi(X) = P_{x,j,k}(t_l)X$. Where X and Y are continuous random variables and f_X and f_Y are their density functions, respectively. Since $\Phi(X)$ is a strictly increasing function on the range of X , using the probability theorem

³ User types are defined from the point of view of the cell, which means that a cell may perceive a user as having a different type than that perceived by another cell.

that says

$$f_Y(y) = f_X(\Phi^{-1}(y)) \frac{d}{dy} \Phi^{-1}(y) \quad (6)$$

we have

$$f_Y(y) = \frac{1}{P_{x,j,k}(t_l)} f_X\left(\frac{y}{P_{x,j,k}(t_l)}\right) \quad (7)$$

Cell k should consider a user x' , for time t_l , with a bandwidth demand distribution function f_Y and use it to perform its local admission control.

We write $r_k(x, t)$ the elementary response of cell k for user x for time t . We assume that $r_k(x, t)$ can take one of two values: -1 meaning that cell k cannot accommodate user x at time t ; and $+1$ otherwise.

The cell determines the order in which it will perform its call admission control for the users. For instance, the cell can sort users in decreasing order of their DMPs.

If we assume that user x_i has a higher priority than user x_j for all $i < j$, then to compute elementary responses for user x_j , we assume that all users x_i with $i < j$ that have a positive elementary response are accepted. As an example, if a cell wants to compute the elementary response r for user x_4 , and we have already computed r for users $x_1=1$, $x_2=1$ and $x_3=-1$, then to compute r for x_4 the cell assumes that users 1 and 2 are accepted in the system but not user x_3 .

6.2. Computing the final responses and sending the results

Since the elementary responses for future foreign users are computed according to local information about the future, they should not be assigned the same confidence as at t_0 .

We denote by $C_k(x, t)$ the confidence that cell k has in its elementary response $r_k(x, t)$. The confidence degrees depend on many parameters. It is clear that the time in the future for which the response is computed has an impact on the confidence in that response. The available bandwidth when computing the elementary response also affects the confidence.

To compute the confidence degrees, we use a formula based on the percentage of available bandwidth when computing the elementary response as an indication of the confidence the cell may have in this elementary response. The confidence degrees are computed using Eq. (8)

$$C_k(x, t) = e^{(p-1)} p^n \quad (8)$$

where p is a real number between 0 and 1 representing the percentage of available bandwidth at the time of computing the elementary response, and $n \geq 1$ is a parameter that is chosen experimentally to obtain the best efficiency of the call admission algorithm.

If, for user x , cell k has a response $r_k(x, t)$ for each t from t_0 to t_{m_x} with a corresponding DMPs $P_{x,j,k}(t_0)$ to $P_{x,j,k}(t_{m_x})$, then to compute the final response those elementary responses are weighted with the corresponding DMPs.

The final response from cell k to cell j concerning user x is then

$$R_k(x) = \frac{\sum_{t=t_0}^{t=t_{m_x}} r_k(x, t)P_{x,j,k}(t)C_k(x, t)}{\sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k}(t)} \quad (9)$$

where $C_k(x, t)$ is the confidence that cell k has about the elementary response $r_k(x, t)$. To normalize the final response, each elementary response is also divided by the sum over time t of the DMPs in cell k . Of course, the sum $\sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k}(t)$ should not be zero (which would mean that all the DMPs for cell k are zero!). Cell k , then, sends the response $R_k(x)$ to the corresponding cell j . Note that $R_k(x)$ is a real number between -1 and 1 .

7. The algorithm

At each time t , cell j decides if it can support new users. It decides locally if it can support users of types 1 and 2, since they have higher priority than other types of user (see the user types in Section 6.1.1). This is because, from a user’s point of view, receiving a busy signal is better than a forced termination. The cell also sends the DMPs to other cells and informs them of its users of type 3. Only those who can be supported locally are included; other users of type 3 that cannot be accommodated locally are rejected. At the same time, the cell receives DMPs from other cells and is informed of users of type 4.

Using Eq. (9), the cell decides if it can support users of type 4 in the future and sends the responses to the corresponding cells. When it receives responses from the other cells concerning its users of type 3, it performs one of the two following steps. If the cell cannot accommodate the call, it is rejected. If the cell can accommodate the call, then the CAC decision depends on the value of $D(x)$ (see Eq. (5)).

7.1. Distributed admission control scheme pseudo code

At time t_0

- (1) Send the DMPs of all type 1 users
- (2) Process type 2 users:
 - Sort these users according to some ordering policy (FIFO, QoS...)
 - Perform a local admission control for each user
 - Send the DMPs of each user accepted
- (3) Process type 3 users:
 - Remove all users that cannot be supported locally
 - Send the DMPs of each user
- (4) Receive DMPs for users of type 4 and for old users in the neighborhood of the cell.
- (5) For $t=t_0 + \tau$ to $MAX_x(t_{m_x})$ do
 - Sort all users according to their DMPs (in a decreasing order)

- (a) Take the user x with highest DMP and who was accepted in the previous step.
- (b) Consider a user x' that has the bandwidth requirement f_Y where f_Y is as in Eq. (7) and f_X is the bandwidth requirement of user x , and process user x' using the local CAC algorithm.
- (c) If user x is of type 3 or 4 then
 - if user x' is accepted, then set $r_j(x, t)$ to 1, else set $r_j(x, t)$ to -1 .
- (d) Compute the confidence degrees $C_j(x, t)$.
- (e) Go to (a) if this is not the last user.
- (6) For all users x_{type4} of type 4, compute the final responses $R_j(x_{type4})$ using Eq. (9).
- (7) Send the results to the corresponding cell (the cell responsible for user x_{type4})
- (8) Receive the final responses for type 3 users x_{type3} and compute the weights using Eqs. (1) and (4) and then compute $D(x_{type3})$.
- (9) For each user x of type 3,
 - if $D(x) \geq T_{acc}$ then user x is accepted, otherwise, the user is rejected.

Fig. 5 depicts the admission process diagram at the cell receiving the admission request and at a cell belonging to the cluster. Because the admission request is time sensitive the cell waiting for responses from the cells in the cluster will wait until a predefined timer has expired then it will

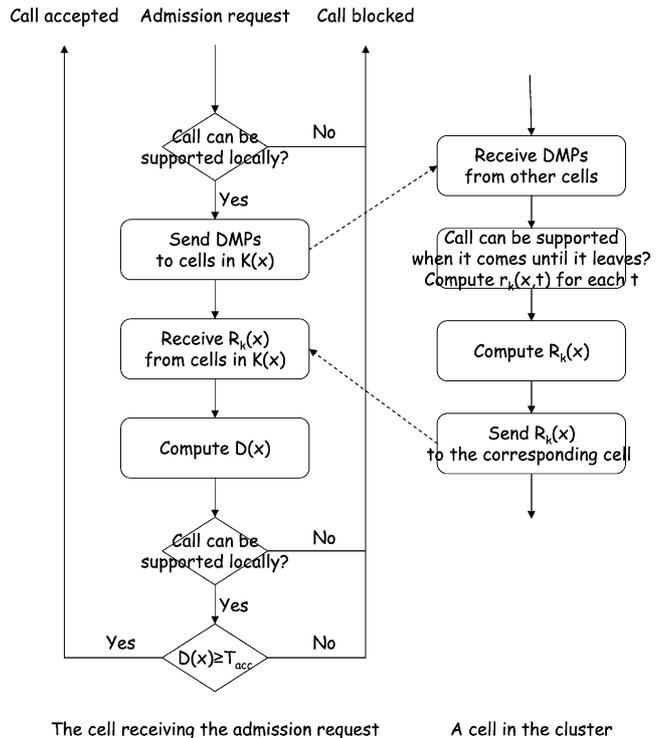


Fig. 5. Admission process diagram.

assume a negative response from all cells that could not respond in time.⁴

8. The efficiency concept

While studying the performance of an admission control algorithm, several performance parameters need to be measured. The commonly measured performance parameters are: call-dropping probability (CDP), average bandwidth utilization (ABU) and call blocking probability (CBP).

Each CAC algorithm in a particular situation can have a particular value for CDP, ABU and CBP. To facilitate the comparison between different CAC schemes, we can represent each CAC scheme as a point (cdp, abu, cbp) in a three-dimensional space with the x -axis indicating the CDP, the y -axis indicating the ABU and the z -axis indicating the CBP.

To compare different CAC schemes we need a reference point. The distance between this reference point and the point indicated by the statistics of a particular algorithm, will determine the performance of the algorithm.

The best possible case is of course the case where the CDP is equal to 0%, the CBP is equal to 0% and the ABU is equal to 100% (i.e. (0, 100, 0%)). However, this case is not realizable as it is not possible to have 100% bandwidth utilization while having a 0% CDP and a 0% CBP. Thus, the best possible case is (min_CDP, max_ABU, min_CBP), where min and max indicate the minimum and the maximum, respectively, over all the algorithms and under the same load. This will be our reference point. The algorithm that has the nearest point to the reference point will have the highest performance. Without loss of generality, we assume that CDP, ABU and CBP have been normalized between 0% and 100% so that the reference point is now (0, 100, 0%).

We define the efficiency of a CAC algorithm as follows

$$E(\text{cdp}, \text{abu}, \text{cbp}) = 1 - \frac{\sqrt{\text{cdp}^2 + (1 - \text{abu})^2 + \text{cbp}^2}}{\sqrt{3}} \quad (10)$$

It is simply, one minus the normalized distance between the reference point and the point indicated by the statistics of the algorithm. Fig. 6 illustrates the 3D space and the concept of efficiency.

9. Simulation model

All the evaluations are done for mobile terminals that are traveling along a highway as in Fig. 2. This is a simple environment representing a 1D cellular system. In our simulation study we have the following simulation parameters and assumptions:⁵

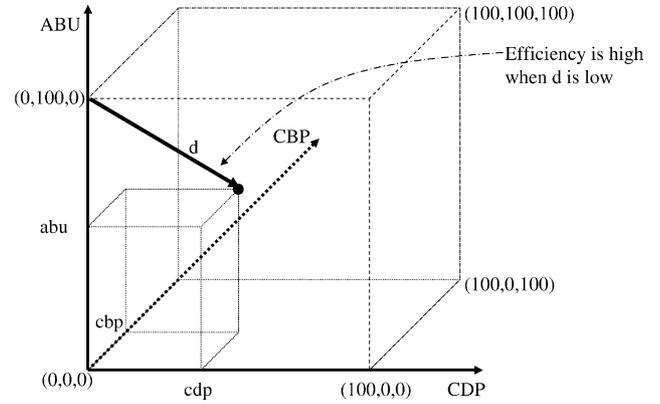


Fig. 6. The efficiency concept.

- (1) The time is quantized in intervals of $\tau = 10$ s.
- (2) The whole cellular system is composed of 10 linearly arranged cells, laid at 1-km intervals. Cells are numbered from 1 to 10.
- (3) Cells 1 and 10 are connected so that the whole cellular system forms a ring architecture as assumed in [14]. This avoids the uneven traffic load that would be experienced by these border cells otherwise.
- (4) Connection requests are generated in each cell according to a Poisson process with rate λ (connections/s). A newly generated mobile terminal can appear anywhere in the cell with equal probability.
- (5) Mobile terminals speeds are uniformly distributed between 80 and 120 km/h, and mobile terminals can travel in either of two directions with equal probability.
- (6) We consider three possible types of traffic: voice, data, and video. The number of bandwidth units (BUs) required by each connection type is: $B_{\text{voice}} = 1$, $B_{\text{data}} = 5$, $B_{\text{video}} = 10$. Note that fixed bandwidth amounts are allocated to users for the sake of comparison with the other algorithms described in Section 9.1. The probabilities associated with voice, data, and video traffic types are $p_{\text{voice}} = 0.3$, $p_{\text{data}} = 0.4$ and $p_{\text{video}} = 0.3$, respectively.
- (7) Connection lifetimes are exponentially distributed with a mean value of 180 s.

For the Distributed Call Admission Control scheme we assume also that:

- (1) The DMPs are computed as in [17].
- (2) The weights are computed using Eqs. (1) and (2).
- (3) The confidence degrees are computed using Eq. (8) with $n = 3$.

Five hours of traffic is simulated in each experiment that has been repeated several times to get results within the 95% confidence interval.

⁴ Alternative behavior can also be adopted.

⁵ The simulation parameters used here are those used by most researchers.

9.1. Simulated admission control algorithms

In addition to the proposed admission control algorithm, we have simulated two other CAC schemes which are the Guard Channel (GC) scheme and the Shadow Cluster (SC) [17] scheme and which are briefly explained below.

In the GC scheme, a number of channels are dedicated in each cell for exclusive use by handoff users. To evaluate this algorithm we simulate a system that uses the GC scheme. We changed the number of reserved channels (from 0 to 100% in steps of 1%) for each simulation and we computed several important QoS statistics.

In the SC scheme, the cell receiving the admission request sends the DMPs to the neighboring cells, however, each of these cells does not send a response for each user, rather it sends a single response per time step (called availability estimates) that indicates the level of congestion of the particular cell. The cell receiving these responses takes an average value (called survivability estimates) and accepts users with a survivability estimate higher than a particular threshold. Note that the SC scheme considers users with fixed bandwidth requirements only.

10. Performance evaluation

We simulated a system that uses our Distributed Call Admission Control scheme, and changed the value of the acceptance threshold (from 0.4 to 0.7 in steps of 0.01) for each simulation and we computed important statistics like the Call Dropping Percentage, the Call Blocking Percentage and the Average Bandwidth Utilization. Also we simulated a system that uses the SC scheme, and changed the value of the admission threshold (from 0.0 to 3.0 in steps of 0.1) for each simulation and we computed the same statistics.

10.1. Simulation scenarios

Several scenarios have been considered and are explained below.

All three algorithms (i.e. DCAC, SC and GC) have been simulated in two situations:

- (1) *No-congestion*. Each cell has a fixed capacity of 100 bandwidth units (or channels).
- (2) *Congestion*. Each cell has a fixed capacity of 100 bandwidth units except cells 3–5 that have 50, 30 and 50 bandwidth units, respectively. This creates a local congestion in the long term. An example of such case is a temporary increase in the interference level that prevents the cells from using all their capacity.

All three algorithms in these two situations (i.e. congestion and no-congestion) have been simulated subjected to the following loads:

Knowing the average connection lifetime and bandwidth, we choose the connection generation rate to have a cell load of 50, 100 and 150. These correspond to normalized loads of 0.5, 1 and 1.5, respectively. The 150 cases have also been simulated with data traffic only.

Note: As in [13], the offered load per cell, L , is defined as connection generation rate \times connection bandwidth \times average connection lifetime, i.e.

$$L = \lambda(p_{\text{voice}}B_{\text{voice}} + p_{\text{data}}B_{\text{data}} + p_{\text{video}}B_{\text{video}})180$$

Also, to investigate the effect of the number of cells of a user cluster, the DCAC and the SC schemes were simulated in the two following cases:

- (1) The size of the cluster $K(x)$ is fixed for all users and is equal to 2. This means that one cell in the direction of the user and the cell where the user resides form the cluster. In this case DCAC and SC are denoted DCAC-k2 and SC-k2, respectively. The value of m_x is fixed for all users and for the duration of the connection and is equal to 18. This means that the DMPs are computed for 18 steps in the future.⁶
- (2) The size of the cluster $K(x)$ is fixed for all users and is equal to 5. In this case DCAC and SC are denoted DCAC-k5 and SC-k5, respectively, and m_x is equal to 26.

The following summarizes the considered simulation scenarios:

- Algorithms: DCAC-k2, DCAC-k5, SC-k2, SC-k5, GC
- Load: 50, 100, 150, 150 with data users only
- State: congestion, no-congestion

10.2. Simulation results: first set

10.2.1. Load is equal to 50

Fig. 7 shows two sets of efficiency curves when the offered load is 50. The first set which contains the five higher curves, represents the results for the five considered algorithms (DCAC-k2, DCAC-k5, SC-k2, SC-k5, GC) in the no-congestion scenario. The second set which contains the five lower curves, represents the results for the five considered algorithms in the congestion scenario. The x -axis represents the achieved CDP, while the y -axis represents the achieved efficiency. The efficiency is computed using Eq. (10).

While the five algorithms achieve almost the same efficiency in the no-congestion scenario, they clearly have different performances in the congestion scenario.

If we read the figure from the right to the left, all the curves (in a particular scenario, i.e. congestion or no-congestion) start from the same point. This point represents the performance achieved when there is no admission control. It is important to

⁶ m_x was chosen to reflect the maximum amount of time needed for a mobile to traverse all cells in the considered cluster.

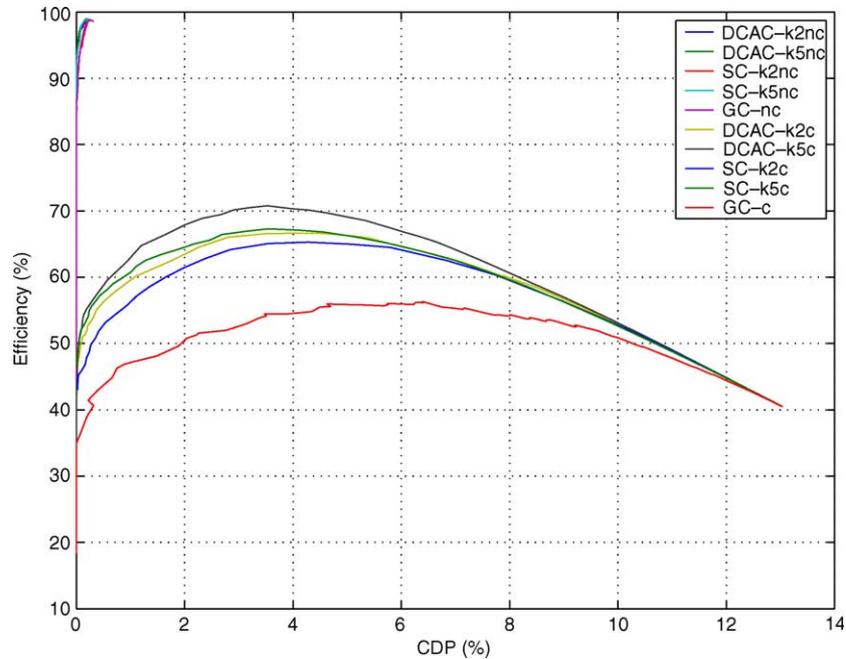


Fig. 7. Efficiency for load=50.

notice that the efficiency starts increasing when we leave this point. This proves that it is worth doing admission control irrespective of the algorithm used. The curves then reach a maximum and then the efficiency starts decreasing as the CDP gets near 0%. We notice that the DCAC and SC schemes have higher efficiency than the GC scheme.

Fig. 8 shows the maximum achieved efficiency by the algorithms in the two considered scenarios. It shows that all the algorithms achieve the same maximum efficiency in

the case of no-congestion. This is because the load in the system is very low. However, in the congestion scenario, the best performance is achieved by the DCAC-k5 scheme. DCAC-k2 and SC-k5 schemes achieve almost the same maximum efficiency, followed by the SC-k2 scheme and then the GC scheme. The DCAC-k5 scheme achieves the highest efficiency because it has the ability to avoid admitting those users who are most likely to be dropped and can use the saved bandwidth to accept more users who can most likely be

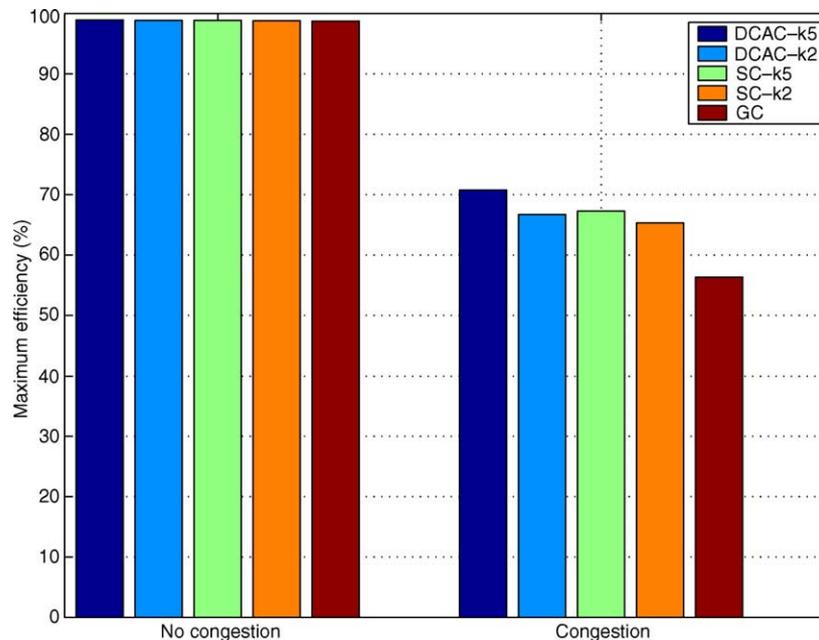


Fig. 8. Maximum efficiency for load=50.

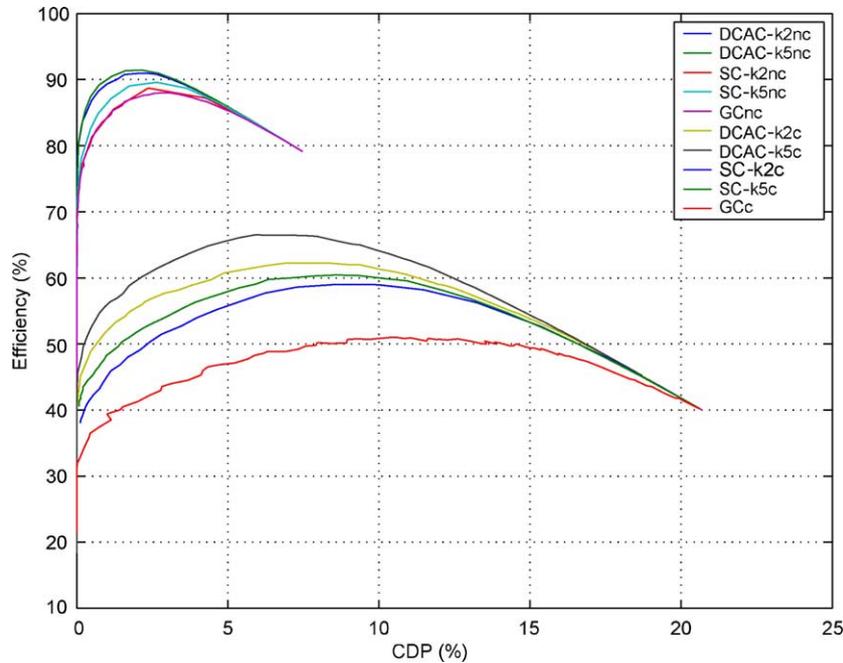


Fig. 9. Efficiency for load = 100.

supported. The results also proves that the weights-allocation strategy allows for better performance results.

10.2.2. Load is equal to 100

Fig. 9 depicts the achieved efficiency when the offered load is 100. The higher curves represent the performance of the five considered algorithms in the no-congestion scenario. The five lower curves represent the performance in the congestion scenario.

In the no-congestion scenario, we can see that the two DCAC schemes achieve almost the same performance. The curve representing the SC-k5 scheme follows with a lower efficiency for all CDP values. Below it, we can notice that the SC-k2 and the GC schemes achieve almost the same performance.

In the congestion scenario, the best performance is achieved by the DCAC-k5 scheme, followed by DCAC-k2, SC-k5 and finally by the SC-k2 and the GC schemes.

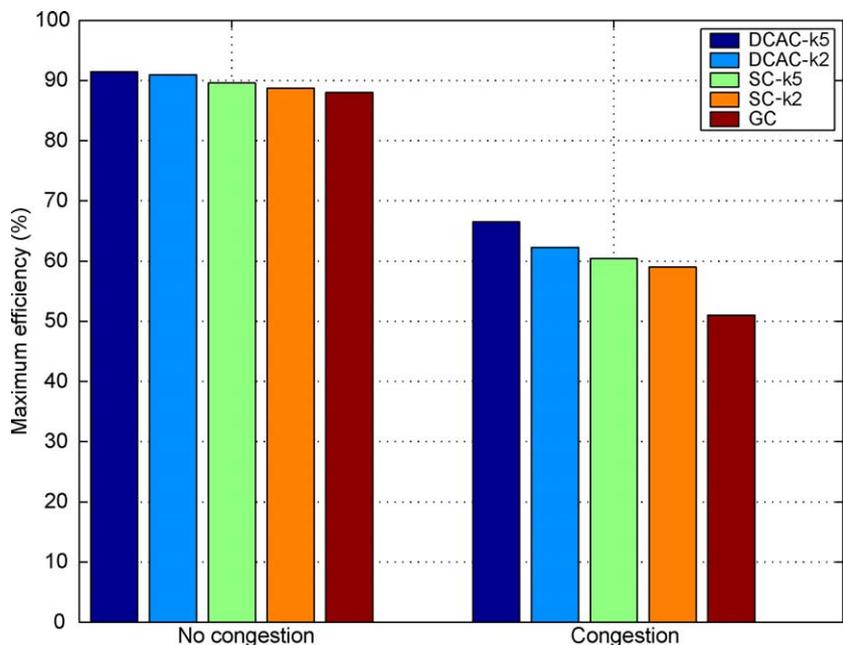


Fig. 10. Maximum efficiency for load = 100.

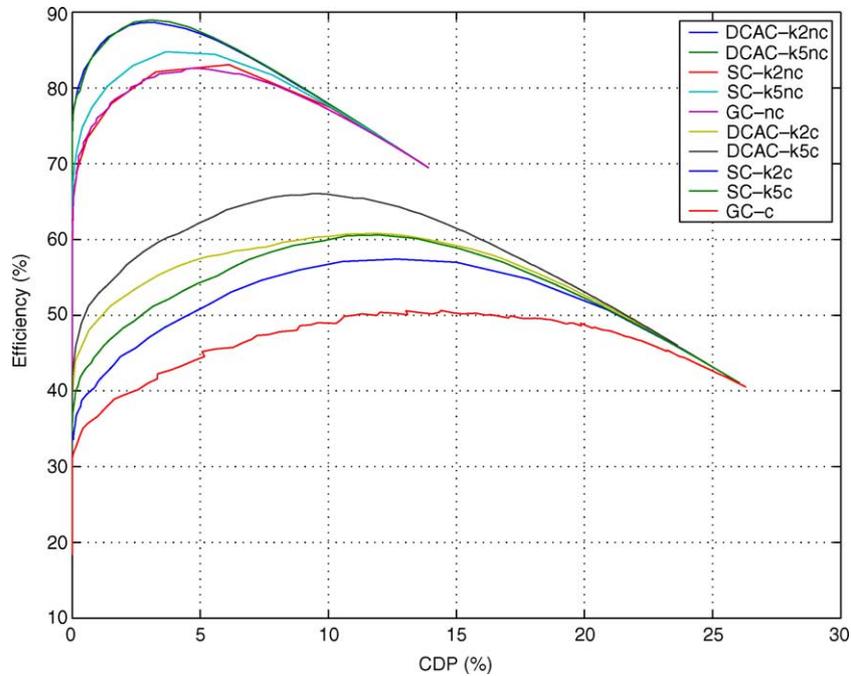


Fig. 11. Efficiency for load=150.

Fig. 10 depicts the maximum achieved efficiency by the algorithms in the two considered scenarios. It shows clearly that the proposed DCAC scheme achieves the best performance. It shows also that the DCAC-k5 scheme performs better than the other algorithms when the load is high.

10.2.3. Load is equal to 150

Fig. 11 depicts the performance of the algorithms when the system is subjected to a load of 150. In this case, the offered load is higher than the capacity of the system.

Here again, in the no-congestion scenario, DCAC-k5 and DCAC-k2 achieve the same performance. The second best performance is achieved by SC-k5 followed by SC-k2 and GC. This is clearly shown in Fig. 12.

In the congestion scenario, DCAC-k5 achieves better performance than SC-k5, and DCAC-k2 achieves better performance than SC-k2. DCAC-k2 and SC-k5 achieve almost the same performance. The worst performance is achieved by the GC scheme.

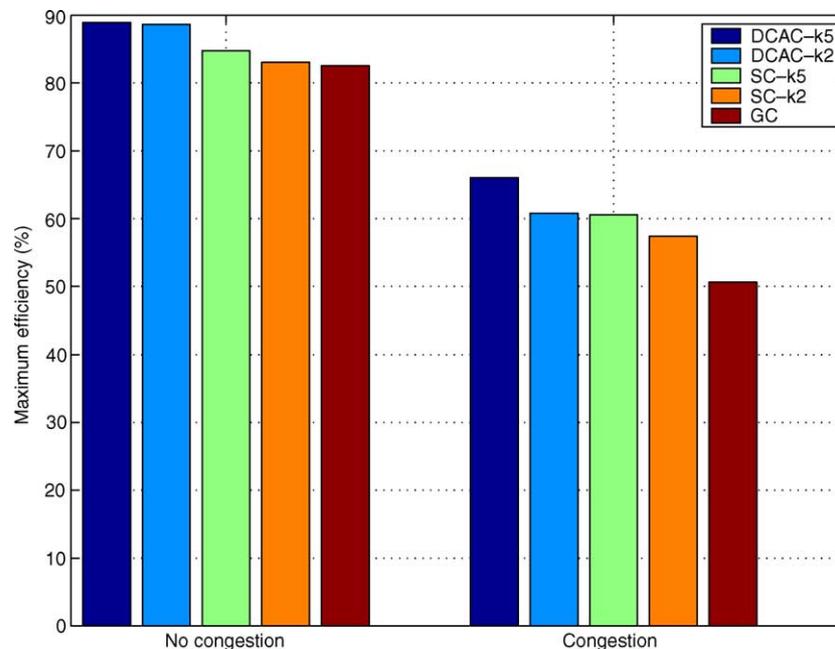


Fig. 12. Maximum efficiency for load=150.

This shows that even without involving several cells, DCAC-k2 is able to differentiate between the users and accept only those that are more likely to finish their calls without being dropped. SC-k5 on the other hand, is not able to make such differentiation even when involving five cells. One reason for this is that in the SC scheme, the cells do not give individual responses about each user, rather they only give an availability estimate that indicates their level of congestion at a particular time. Another important reason is the lack of judicious weight allocation strategy in the case of SC.

DCAC-k5 takes advantage from both involving more cells in the admission decision and the individual responses from those cells. Also, the unique way DCAC combines the responses from the cells involved in the CAC process, allows it to take more clear-sighted decisions and to achieve higher efficiency.

The inferior performance achieved by DCAC-k2 in comparison to DCAC-k5 is explained by its smaller cluster that prevents the scheme from being informed of and taking into account a distant congestion. Thus, the only way for DCAC-k2 to reduce the CDP is to accept fewer users, which results in poorer resource utilization.

On the other hand, since DCAC-k5 involves more cells in the CAC process than DCAC-k2, the scheme is able to distinguish between those users who can be supported and those who are most likely to be dropped due to congestion. This has the two following benefits: (1) the scheme can accept more users without sacrificing the CDP, and (2) the bandwidth saved by not allowing some ‘bad’ users to be admitted can be used to admit more ‘good’ users, particularly in heavy load situations.

10.2.4. Load is equal to 150 with data users only

In this simulation set, we consider data traffic only. This is to make all users having the same bandwidth requirements and looking at the performance of the considered algorithms. This tests the ability of the considered algorithms to differentiate between the users and accepting only those that can be supported while achieving high efficiency.

Table 3 shows the maximum achieved efficiency by the five algorithms in the two scenarios (i.e. no-congestion and congestion).

According to the table, all five algorithms achieve almost the same maximum of efficiency in the no-congestion scenario. This is because the load is uniform. The fact that

Table 3
Maximum efficiency for load=150 (data users only)

	No-congestion (%)	Congestion (%)
DCAC-k5	85.2528	64.5164
DCAC-k2	84.8581	56.6152
SC-k5	84.8385	58.5847
SC-k2	84.3619	55.6119
GC	84.1442	49.4571

the DCAC-k5 and DCAC-k2 schemes achieve almost the same performance in a network with uniformly distributed load is intuitively predictable. This is mainly because the responses from the three additional cells in DCAC-k5 (cells 2–4 in Fig. 2) only confirm what the two cells in DCAC-k2 (cells 0, 1 in Fig. 2) have decided.

In the congestion scenario, DCAC-k5 clearly outperforms the other algorithms. This shows that the DCAC scheme is able to differentiate between the users even if they have the same bandwidth requirements. DCAC-k2 has a slightly higher efficiency than SC-k2 and not very far from the performance achieved by SC-k5.

We have conducted several other simulations with different offered loads and different simulation parameters. Besides the fact that DCAC-k5 outperforms all the other schemes in all situations, the main observation worth highlighting here is that the two schemes DCAC-k5 and DCAC-k2 achieve almost the same performance in the case of no-congestion or of uniformly distributed congestion. The latter case is less important since it can be solved off-line by increasing the network capacity. We have observed in the simulations that DCAC-k5 achieves better performance in case of local congestion.

Of course, DCAC-k5 does have some disadvantages. As DCAC-k5 involves more cells in the CAC decision process, it induces more communications between base stations and requires more processing power than DCAC-k2. These resources are less critical compared to the wireless network bandwidth. A good compromise is to use DCAC-k2 when the network is not congested and use DCAC-k5 when congestion is detected. The process of selecting the best scheme is out of the scope of this paper and is a subject for future work.

Also, the efficiency curves suggest that rather than changing the acceptance threshold to have a particular CDP, it is better to change the acceptance threshold to achieve a higher efficiency.

10.3. Simulation results: second set

In this section, we investigate the impact of changing some of the parameters on the performance of the considered algorithms. In particular, we investigate the effect of the mean holding time and the shape of the cells.

10.3.1. The effect of the mean holding time

In this case, we want to investigate the effect of the mean holding time on the achieved performance. The considered algorithms in this case are: DCAC-k2, SC-k2 and GC.

All simulation parameters are as in Sections 9 and 10.1 except the followings:

- No-congestion. Each cell has a fixed capacity of 100 bandwidth units (or channels).
- The load is equal to 150 which corresponds to a normalized load of 1.5.

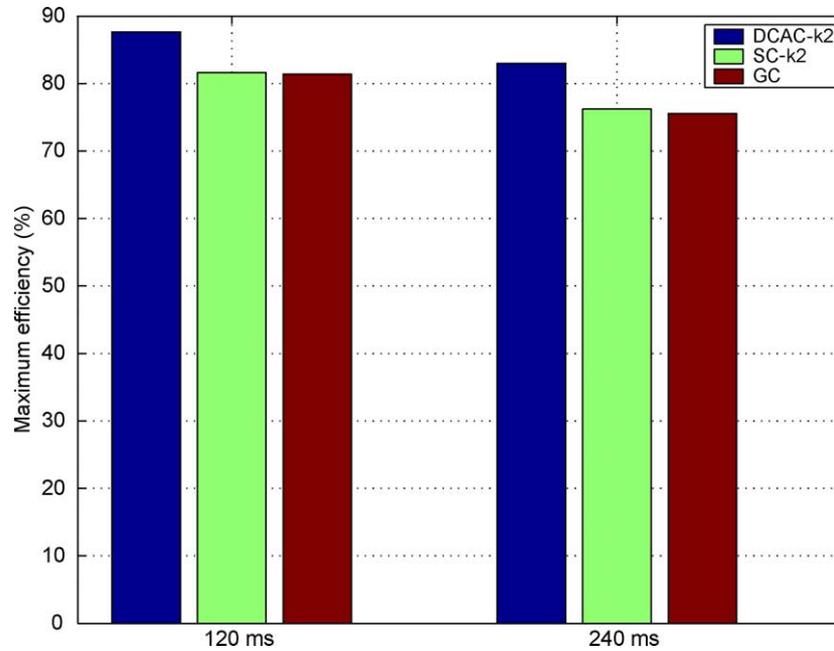


Fig. 13. Maximum efficiency for the two considered mean holding times.

The three considered algorithms were simulated in the two following cases:

- (1) Connection lifetimes are exponentially distributed with a mean value of 120 s.
- (2) Connection lifetimes are exponentially distributed with a mean value of 240 s.

Fig. 13 depicts the obtained results. The figure shows the maximum efficiency attained by the three considered algorithms in the two scenarios. It is clear from the figure that the relative merit of the three algorithms does not change. In other words, the proposed DCAC-k2 performs better than SC-k2 and GC irrespective of the mean holding time. However, as expected, the performance of each algorithm decreases as the mean holding time increases. This is intuitively expected. As users spend more time in the network with a longer mean holding time, they have more chance to get dropped and hence the achieved dropping probability is higher than the case where the mean holding time is smaller. To achieve the same dropping probability, each algorithm needs to perform more bandwidth reservation which results in lower efficiency.

10.3.2. The effect of the shape of the cells

In this set of simulations, we want to investigate the effect of the shape of the cells on the achieved performance. The considered algorithms in this case are: DCAC-k2, SC-k2 and GC.

All simulation parameters are as in Sections 9 and 10.1 except the followings:

- Congestion. Each cell has a fixed capacity of 100 bandwidth units except cells 3–5 that have 50, 30 and 50

bandwidth units, respectively. This creates a local congestion in the long term.

- The load is equal to 150 which corresponds to a normalized load of 1.5.
- Connection lifetimes are exponentially distributed with a mean value of 240 s.

The three considered algorithms were simulated in the two following cases:

- (1) Handoffs are executed at the border of adjacent cells.
- (2) The shape of the cells are fuzzy and the handoff between cells can occur randomly anywhere within 10% of the border in each side. This case is more realistic as the coverage of the cells changes over time. This will have an impact on the prediction of user movements.

Table 4 depicts the obtained results. The table shows the maximum efficiency attained by the three considered algorithms in the two scenarios. Here again, it is clear from the table that the relative merit of the three algorithms does not change, i.e. the proposed DCAC-k2 performs better than SC-k2 and GC. However, in this case, the difference in the achieved performance between the case of fixed border and that of fuzzy border is not as remarkable as the effect of the mean holding time. The simulations were repeated with various parameters and

Table 4
Maximum efficiency for fixed and fuzzy cell border

	DCAC-k2 (%)	SC-k2 (%)	GC (%)
Fixed border	65.0361	61.4624	54.2598
Fuzzy border	64.3921	60.9067	53.7571

similar results were obtained, mainly there is no significant impact of the shape of the cells on the performance of the considered algorithms. The maximum difference in the efficiency is about 1%. This is true only for the considered square-shaped cells. This can be explained by the fact that the generated error, due to the existence of the fuzzy border between the cells, is averaged over all users. More specifically, if a cell predicts that a user will stay in the cell and that another will leave the cell, if these two predictions are false, the effect of these two errors (in average and over all the system) will be balanced.

11. Conclusion

In this paper, we have identified two important parameters that have to be taken into account when combining the responses from the different cells involved in the admission process. To each cell we have associated a *temporal* and a *spatial* relevance. The temporal relevance indicates the importance of the response of the cell based on the percentage of time the user may spend in this cell in comparison to the time the user may spend in the whole cluster. The spatial relevance indicates the importance of the response of the cell based on its position within the cluster. These two parameters allow to better evaluate the contribution of each cell in the final decision regarding the admission of a new user.

We have also presented a call-admission-control scheme tailored for wireless multimedia networks. The scheme operates in a distributed fashion by involving, in the admission process, not only the cell where the call originates, but also a certain number of neighboring cells. The novel way of combining the responses of neighboring cells allows the proposed scheme to make better decisions and hence to achieve higher efficiency.

We have also introduced a new combined QoS metric for comparing the performance of admission control algorithms. This new metric called ‘efficiency’ combines the three most important performance parameters that are the call-dropping probability, the call blocking probability and the average bandwidth utilization.

Simulations results have shown a significant improvement of the distributed CAC over the Guard Channel and the Shadow Cluster schemes. By implementing the distributed call admission control scheme and the weight allocation strategy, the system is able to reduce the call-dropping probability while achieving high resource utilization. This is achieved by selecting those users that are more likely to complete their call without being dropped.

We also presented an analysis of the comparison between call admission control schemes involving different numbers of cells in the decision process. We observed that it is worthwhile to involve more cells in the CAC decision in the case of local congestion. However, the performance may be

poor even if we involve more cells in the admission control process if the weight allocation is not done properly. The choice of the optimal number of cells to involve and when this should happen is an important issue that will be investigated in the future.

References

- [1] Universal Mobile Telecommunications System (UMTS), <http://www.etsi.org/umts/>.
- [2] T. Zhang, E.v.d. Berg, J. Chennikara, P. Agrawal, J.-C. Chen, T. Kodama, Local predictive resource reservation for handoff in multimedia wireless IP networks, *IEEE Journal on Selected Areas in Communications (JSAC)* 19 (10) (2001) 1931–1941.
- [3] Y. Fang, Y. Zhang, Call admission control schemes and performance analysis in wireless mobile networks, *IEEE Transactions on Vehicular Technology* 51 (2) (2002) 371–382.
- [4] S. Wu, K.Y.M. Wong, B. Li, A dynamic call admission policy with precision QoS guarantee using stochastic control for mobile wireless networks, *IEEE/ACM Transactions on Networking* 10 (2) (2002) 257–271.
- [5] C.-T. Chou, K.G. Shin, Analysis of combined adaptive bandwidth allocation and admission control in wireless networks, in: *IEEE Conference on Computer Communications (INFOCOM)*, 2002 pp. 676–684.
- [6] C.W. Ahn, R.S. Ramakrishna, QoS provisioning dynamic connection-admission control for multimedia wireless networks using a hopfield neural network, *IEEE Transactions on Vehicular Technology* 53 (1) (2004) 106–117.
- [7] D. Hong, S. Rappaport, Traffic modeling and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized handoff procedures, *IEEE Transactions on Vehicular Technology* 35 (1986) 77–92.
- [8] R. Guerin, Queuing blocking system with two arrival streams and guard channels, *IEEE Transactions on Communications* 36 (1988) 153–163.
- [9] D.A. Levine, I.F. Akyildiz, M. Naghshineh, The shadow cluster concept for resource allocation and call admission in ATM-based wireless networks, *ACM/IEEE International Conference on Mobile Computing and Networking*, 1995, pp. 142–150.
- [10] S. Lu, V. Bharghavan, Adaptive resource management algorithms for indoor mobile computing environments, *ACM Special Interest Group on Data Communication (SIGCOMM)*, 1996, pp. 231–242.
- [11] A. Aljadhari, T.F. Znati, A framework for call admission control and QoS support in wireless environments, in: *IEEE Conference on Computer Communications (INFOCOM)*, vol. 3 1999, pp. 1019–1026.
- [12] A. Aljadhari, T.F. Znati, Predictive mobility support for QoS provisioning in mobile wireless networks, *IEEE Journal on Selected Areas in Communications (JSAC)* 19 (10) (2001) 1915–1930.
- [13] S. Choi, K.G. Shin, Adaptive bandwidth reservation and admission control in QoS-sensitive cellular networks, *IEEE Transactions on Parallel and Distributed Systems* 13 (9) (2002) 882–897.
- [14] M. Naghshineh, M. Schwartz, Distributed call admission control in mobile/wireless networks, *IEEE Journal on Selected Areas in Communications (JSAC)* 14 (4) (1996) 711–717.
- [15] S. Choi, K.G. Shin, Predictive and adaptive bandwidth reservation for handoffs in QoS-sensitive cellular networks, *ACM Special Interest Group on Data Communication (SIGCOMM)*, 1998, pp. 155–166.
- [16] C. Oliveira, J.B. Kim, T. Suda, An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks, *IEEE Journal on Selected Areas in Communications (JSAC)* 16 (6) (1998) 858–874.

- [17] D.A. Levine, I.F. Akyildiz, M. Naghshineh, A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept, *IEEE/ACM Transactions on Networking* 5 (1) (1997) 1–12.
- [18] R.G. Fry, A. Jamalipour, A precision prediction call admission control in packet switched multi-service wireless cellular networks, *IEEE Global Communications Conference (GLOBECOM)*, 2003, pp. 4122–4126.
- [19] B.M. Epstein, M. Schwartz, Predictive QoS-based admission control for multiclass traffic in cellular wireless networks, *IEEE Journal on Selected Areas in Communications (JSAC)* 18 (3) (2000) 523–534.
- [20] B. Eklundh, Channel utilization and blocking probability in a cellular mobile telephone system with directed retry, *IEEE Transactions on Communications* 34 (4) (1986) 329–337.
- [21] H. Wellenhopf, B.H. Lichtenegger, J. Collins, *GPS: Theory and Practice*, Springer, New York, 1994.