Connection admission control in the UTRAN transport network

Szilveszter Nádas, Sándor Rácz, Szabolcs Malomsoky, and Sándor Molnár

Sz. Nádas, S. Rácz, and Sz. Malomsoky are with the Traffic Analysis and Network Performance Laboratory, Ericsson Research, S. Molnár is with the High Speed Networks Laboratory, Dept. of Telecommunications and Media Informatics, Budapest University of Technology and Economics.

The contact person is Sz. Nádas; tel: +36-1-4377382, e-mail: szilveszter.nadas@ericsson.com, fax: +36-1-4377767, address: Traffic Analysis and Network Performance Laboratory, Ericsson Research, Laborc u. 1., Budapest 1037, Hungary.

Abstract

In this paper we propose a Connection Admission Control (CAC) algorithm to provide Quality of Service (QoS) in UMTS Terrestrial Radio Access Networks (UTRAN). In UTRAN, the main QoS requirement is to ensure low packet delays. The CAC algorithm works with priority scheduling, which is used for QoS differentiation. We give a detailed investigation on the performance implications of applying priority scheduler. We provide novel closed-form formulae which are fast, simple and accurate enough for practical implementation of the CAC. A comprehensive performance evaluation study with illustrative numerical examples is also presented. The results are validated by simulations.

Index Terms

UTRAN, connection admission control, QoS differentiation, Gaussian approximation

I. INTRODUCTION

In the transport network layer of UMTS Terrestrial Radio Access Networks (UTRAN), either Internet Protocol (IP) technologies [2] or the Asynchronous Transfer Mode (ATM) in combination with the ATM Adaptation Layer type 2 (AAL2) [1], [3] are used to transport radio frames over the Iub interface, which connects radio network controllers (RNC) and base stations. Figure 1 presents the QoS model of UMTS. If the quality of service (QoS) requirements are met at each service component (e.g., radio bearer service, Iu bearer service, backbone bearer service, etc.), then the end-to-end QoS requirements can also be satisfied. How to provide QoS using admission control over the Wide-band Code Division Multiple Access (WCDMA) radio interface has been studied extensively [12], [13], [14], [15]. In this paper we will focus on the Iub/Iur transport network service (presented with bold characters in Figure 1), where QoS differentiation is handled by scheduling and QoS is guaranteed by connection admission control (CAC). The principles of the CAC algorithm in the transport network can be the same for both transport technology options (IP and ATM/AAL2). The CAC algorithm discussed in this paper runs at the RNC (in downlink), the base station (in uplink), and at each traffic concentrator node (e.g., AAL2 switch).

Since packet delays depend on the scheduling principle applied, a different CAC algorithm is needed for each scheduling method. Scheduling methods have a considerable influence on the achievable link utilization. In references [17], [21], [22] and [24], the efficiency of different scheduling algorithms in UTRAN are compared. With the objective of maximizing link

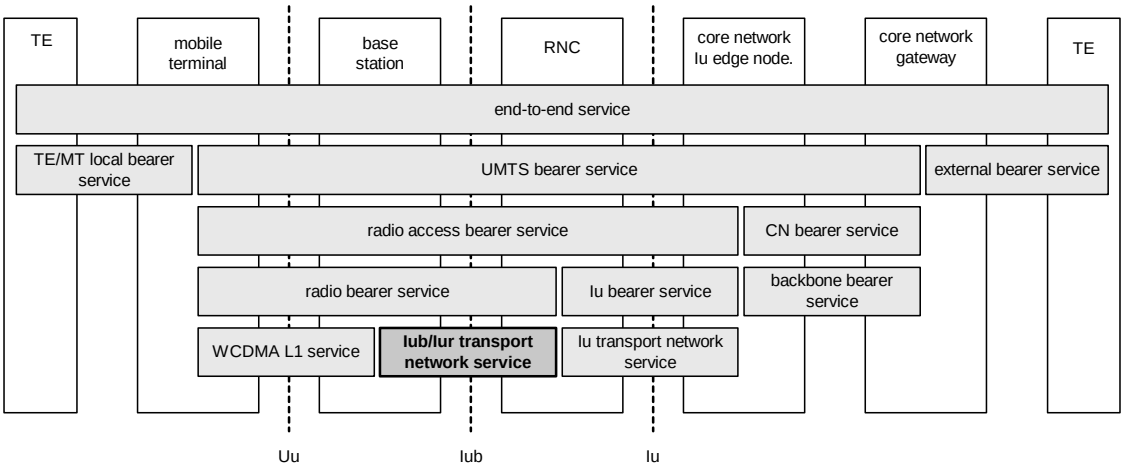

Fig. 1. The QoS model of UMTS [4] (TE - terminal equipment, MT - mobile terminal, RNC - radio network controller, CN - core network)

utilization, the first and second paper proposes the WRR (Weighted Round Robin) scheduling, the third proposes a modified version of the EDF (Earliest Deadline First) algorithm, while the fourth applies class-based WFQ (Weighted Fair Queueing). Among these papers, only [24] proposes also a CAC algorithm, which is, however, not applicable in practice due to computational complexity. All papers point out that QoS differentiation is worth doing in UTRAN. A possible way of solving QoS differentiation is to apply priority scheduling, which is investigated in the present paper. Using priority scheduling, high utilization can also be achieved, because the difference in delay requirements of the various connection types is typically large enough (see [17], [22]). Additional advantages of priority scheduling include that it does not need parameter setting as WRR or EDF, and it is simple to implement.

A summary of earlier work on queueing systems with priority scheduling can be found in [19]. An important development is presented in [26], where the concept of effective bandwidths (see e.g., [28]) is extended for priority scheduling. An accurate analytical method for calculating queueing delays in UTRAN with strict priority scheduling is presented in [23]. In spite of its complexity, this method has been used in the CAC algorithm of [24].

Simplicity and computational efficiency is a determining factor when deciding which CAC algorithm to use, therefore in the present paper we rely on approximations. We use the empty buffer approximation, which was also used in [26] and [27] to obtain workload bounds for priority systems, and develop further the work in [16] by extending it for the case of priority scheduling. We note that the methods presented in [24] can also benefit from our

investigations.

In order to help understanding the task of the CAC, we need to provide relevant details on how UTRAN works [7]. UTRAN operates with so called radio bearers (RBs), which are packet switched radio connections with dedicated resources. CAC decides whether a newly arrived RB connection can be accepted in such a way that the packet delay requirements of all connections in the transport network are met. It makes its decisions based on traffic descriptors and QoS parameters.

The periodic ON-OFF model is suitable for describing the traffic of a connection [16], [18], [21], [22]. The inter-arrival time of packets on a connection is constant, because the medium access control (MAC) protocol schedules them periodically according to the timing requirements of the WCDMA radio interface. This period is called transmission time interval (TTI). In the model, one connection is composed of ON and OFF periods, i.e. time intervals, when a packet is sent within each TTI, and intervals, when packets are not sent. Packet sending times of different connections are independent, because the phases of packet sending on different connections are randomly distributed over the TTI [5] to reduce the probability of packet congestion in the buffers of Iub links.

The parameters of the above model are the following. We denote the number of traffic classes by K, and describe the traffic of a class i connection by three parameters: the packet size (b_i), the packet inter-arrival time (TTI_i), and the activity factor (α_i), which is a number between 0 and 1, and is defined as the average length of ON periods divided by the sum of the average lengths of ON and OFF periods. Using such a simple model ensures that the admission control problem remains tractable such that the usability of the resulting CAC algorithm is not limited.

The CAC must ensure that the probability of a class i packet being delayed by more than its target maximum delay ($\tilde{D}_i$) is kept below a small target value, $\tilde{\varepsilon}_i : \Pr\{D_i > \tilde{D}_i\} \leq \tilde{\varepsilon}_i$, where D_i represents the delay of an arbitrary packet from class i. The delay requirement of a connection is typically smaller than the TTI of the RB carrying the connection ($\tilde{D}_i \leq TTI_i$). For voice traffic (with 20 ms TTI) the delay budget within UTRAN is around 5-7 ms [10]. For data traffic the delay should also be kept low, but somewhat larger delays are tolerated (10-15 ms). Note that these QoS targets for data traffic are not dictated directly by user requirements, but are determined by the requirements of the upper layer, which does the frame synchronization between the base station and the RNC [5]. In downlink, this means that a frame in the RNC need to be transmitted with a certain offset in order to make sure

that the content of the frame arrives in time to the base stations for transmission over the air. Due to soft handover, copies of the same frame may need to be available at several base stations for synchronized transmission over the air.

Because of the strict delay requirements the activity factor sufficiently characterizes the ON-OFF behavior. According to the UTRAN system model, the CAC needs to consider a queueing model that is accurate if the delay requirements are strict (5-15 ms) and the buffer size is small (e.g., smaller than 20 ms). The ON and OFF periods are bursty, meaning that typically both are many TTI long. However, the long term correlation characteristics of the arrival process can not be easily taken into account, because traffic descriptors do not contain any information on the correlation structure of the sources (see the parameters of the Variable Bandwidth Stringent Transfer Capability in [1]), and it is also difficult to get information on this by measurements. This is not a problem in practice, because the buffer is small enough such that it fills up quickly even if so many connections are temporarily in ON state at the same time, that the server can not serve within one TTI the packets arriving in one TTI. Therefore we can assume that the ON-OFF burst component of the queue is negligible and use the approximation that all packets arriving during a temporary overload situation violate the delay requirement (more details are presented in [16]).

Packet sizes and TTI values are RB specific, and can be determined for example from 3GPP standards [6], [7], [8] (we use the values presented in Table II).

Activity factors depend on user behavior and RB allocation procedures. Figure 2 shows a voice call that uses three radio cells during its lifetime, thus it is transferred in three RB connections over different Iub links and consequently handled by three different CAC entities. We can see that the activity factors on these connections depend on the way the subscriber speaks and on how the codec works. However, for RB connections, which are set up during data sessions the activity factor depends more on system behavior, and less on user behavior. As indicated in Figure 3, for data sessions channel type switching is used [9], which means that small amounts of data are transmitted on common channels (e.g., on FACH - forward access channel), and if larger amount of data is to be transmitted, a dedicated channel (DCH) is set up. At the end of data bursts the DCH is teared down. For DCHs RB connections are set up dynamically, therefore high activity factors are expected on these connections. In this paper we do not study the proper setting of the activity factor. In the numerical examples, we will use the activity factor 1 for data traffic (in other words, we assume ideal channel type switching). However, note that the proposed CAC algorithm works for any activity factor

setting.

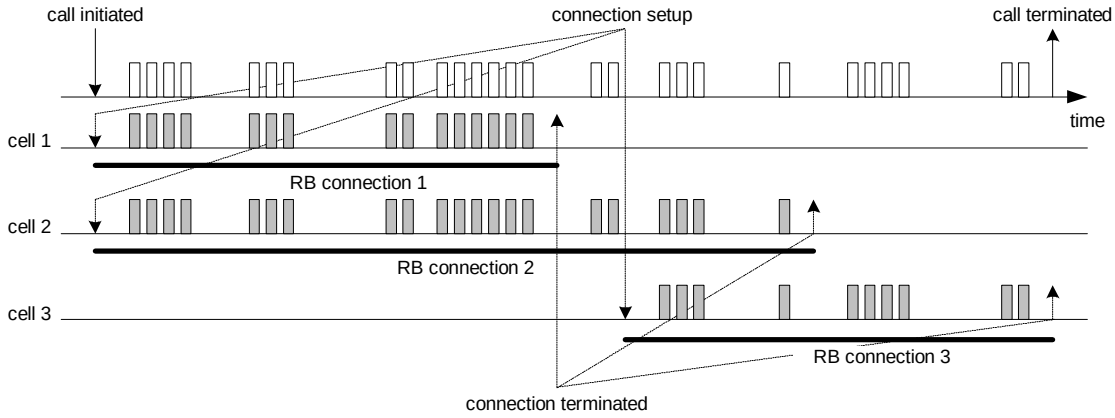

Fig. 2. RB connections during a voice call

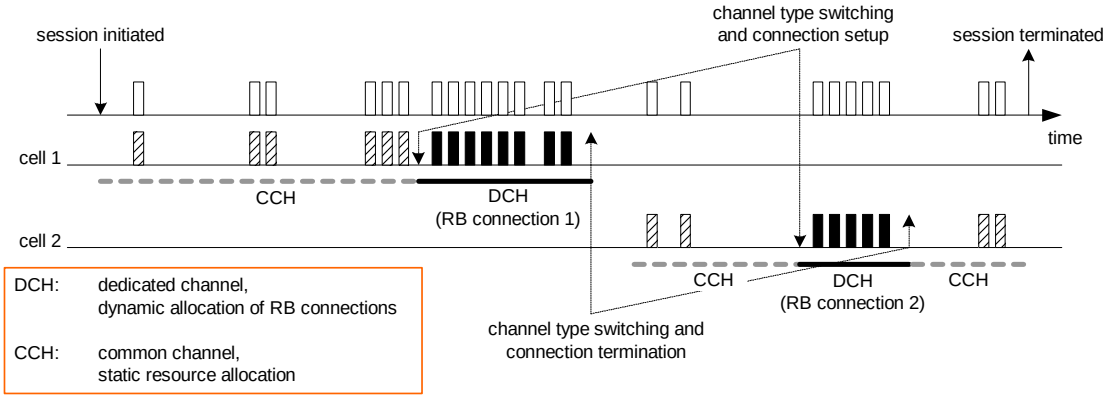

Fig. 3. RB connections during a data session are set up only for dedicated channels

Summarizing our main assumptions, we consider priority scheduling, the RB connections are modeled by independent periodic ON-OFF sources, where the ON-OFF behavior is described by the activity factor only, delay requirements are strict and buffers are small. In order to obtain a simple but accurate algorithm, the packet-level and the burst-level queueing phenomena are considered separately (see [16] and the next section), continuous approximation of the arrival process is used, and the empty buffer approximation is applied. The implications on validity of results of these assumptions will mainly be discussed at the end of the paper, where we show that the CAC algorithm is conservative and reaches high link utilization.

Our main contributions are the following:

- We have extended the algorithm presented in [16] for strict priority scheduling and derived efficient closed-form formulas that can be used in the CAC for QoS provisioning.
- We have made a performance evaluation of CAC with different calculation alternatives and validated the results by simulations.
- We have analyzed the applicability of priority scheduling vs. FIFO in the UTRAN transport network.

The paper is organized as follows. In Section II the queueing model is established. The CAC algorithm is presented in Section III. Section IV describes novel closed-form formulae, which are used in the CAC for QoS provisioning. Numerical examples are given in Section V, where we study the accuracy of the proposed CAC and demonstrate the applicability of priority scheduling in UTRAN. The paper is concluded in Section VI.

II. QUEUEING MODEL

In this section a model is provided which allows us to derive the probability of the packet delay criterion violation: $\Pr\{D_i > \tilde{D}_i\}$. Using the terminology of queueing systems, our model corresponds to a modulated $\sum N_i \mathbf{D}_i^{\mathbf{X}_i}/\mathbf{D}/1$ system with strict, non-preemptive priority scheduling, where $\mathbf{X}_i$ refers to batch size and $\mathbf{D}$ refers to deterministic inter-arrival and service time. There are N_i independent class i connections in the system. Connections within the same class are characterized by the traffic descriptors $\{b_i,\ TTI_i,\ \alpha_i\}$ and the priority level p_i (smaller p_i means higher priority). The server capacity is denoted by C. Incoming packets are segmented into segments of size s. If we were interested in FIFO scheduling, references [19] and [20] would provide us with the solution of this queueing model. However, the computation provided there could not be used in a CAC, because it can not be performed in real-time.

Delay criterion violations are caused by two effects:

- the ON-OFF behavior, which results in temporary overloads.
- even if the system is not overloaded, due to the random assignment of connections to transmission phases, the superimposed packet arrivals can also result in packet congestion.

The consequence of strict delay requirements is that the delay quickly reaches the predefined delay criteria given an overload situation, where the queue continuously grows (i.e., when the queue is instable for several TTIs, because too many connections are in ON state). Therefore, the queue cannot cope with an overload situation efficiently even if considering

infinite buffer. We consider systems where buffer sizes are finite, longer than the delay criteria, but short enough to ensure that packets, which suffered too much delay mostly get dropped. We assume that in an overload situation the delay of all packets is always larger than the delay criteria, and that in a non-overload situation the system is emptied at least once within the largest TTI denoted by TTI^{max} (for a more detailed explanation see [16]).

Applying the assumptions above we set up a combined model: we observe two types of packet delay criterion violation events. The first type of this event occurs when the system is overloaded. In this case the buffer is almost always full, therefore incoming packets will be lost or suffer high delays. The second type of this event occurs when the system is not overloaded, but packets are still delayed for longer than their delay criteria. We define two measures as follows: $\epsilon_i^{overload}$, the fraction of class i packets arriving into overloaded system, and $\epsilon_i^{delayed}$, the fraction of class i packets, which arrive into a non-overloaded system, but still are delayed.

Denote the number of active connections (the connections in ON state) at time t of class i by $N_i^{act}(t)$ and let $\underline{N}^{act}(t) = [N_1^{act}(t), N_2^{act}(t), \ldots, N_K^{act}(t)]$. At a time t_0, we say that the system is in state $\underline{n}$ if the random vector $\underline{N}^{act}(t_0)$ takes the value $\underline{n}$ (i.e., $N_i^{act}(t_0) = n_i$; $i = 1, 2, \ldots, K$). The steady-state probability of state $\underline{n}$, denoted by $\Pi(\underline{n})$, is given by a multi-dimensional binomial distribution as follows:

$$\Pi(\underline{n}) = \prod_{i=1}^{K} \Pi_i(n_i) = \prod_{i=1}^{K} \binom{N_i}{n_i} \alpha_i^{n_i} \ (1-\alpha_i)^{N_i - n_i}, \tag{1}$$

where $\Pi_i(n_i) = \Pr\{N_i^{act}(t_0) = n_i\}$.

Since the segment size (s) is typically small, the high priority queues are hardly affected by low priority traffic. Therefore, when checking whether the buffer of priority level p_i is overloaded, the input rate can be calculated as: $R(p_i, \underline{n}) = \sum_{j=1}^{K} \mathcal{I}_{\{p_j \le p_i\}} \ n_j \ \rho_j$, where $\rho_i = b_i / TTI_i$ is the rate of an active connection, and $\mathcal{I}_{\{expression\}}$ is the indicator function. The measure $\epsilon_i^{overload}$ is approximated by the probability that a packet of class i arrives at an overload situation (when $R(p_i, \underline{n}) > C$):

$$\epsilon_i^{overload} \approx \varepsilon_i^{overload} = \Pr\{\text{packet arrives at overload}\} = \frac{\sum_{\underline{n}: R(p_i,\underline{n}) > C} \ n_i \ \Pi(\underline{n})}{\sum_{\forall \underline{n}} n_i \ \Pi(\underline{n})}. \tag{2}$$

In the case of a non-overload situation ($R(p_i, \underline{n}) \le C$), the waiting time is determined by the periodic packet emissions. The measure $\epsilon_i^{delayed}$ is approximated by the probability that a packet arriving in a non-overload situation is *delayed*:

$$\epsilon_i^{delayed} \approx \varepsilon_i^{delayed} = \frac{\sum_{\underline{n}: R(p_i,\underline{n}) \le C} \ n_i \ \Pi(\underline{n}) \cdot \Pr\{D_i > \tilde{D}_i \mid \underline{N}^{act}(t_0) = \underline{n}\}}{\sum_{\forall \underline{n}} n_i \ \Pi(\underline{n})}. \tag{3}$$

The probability of delay criterion violation is the sum of two probabilities: $\varepsilon_i = \varepsilon_i^{overload} + \varepsilon_i^{delayed}$. (Delays of lost packets are considered to be infinite.)

The above model can also be efficiently simulated. We need to consider only the random phase selection and the random selection of the ON and OFF states, but not the correlation structure of the ON and OFF periods. A simulation cycle is run for the $[0, 3\ TTI^{max}]$ interval, and the delay of packets arriving in the $[TTI^{max}, 2\ TTI^{max})$ interval is measured where the system is already in steady state. (A packet arriving at the end of the second TTI^{max} period may be served only in the third period.) If in the measured interval, more packets arrive than can be served, then delays are considered to be infinite. At the beginning of a cycle the state and the phase of each connection are determined. For a class i connection, the state is ON during the whole cycle with probability α_i and the sending phase is distributed according to the uniform distribution over $[0, TTI_i)$. Repeating the above cycles, ε_i can be obtained. Except for the next paragraph, in this paper "simulation" means the simulation procedure described above.

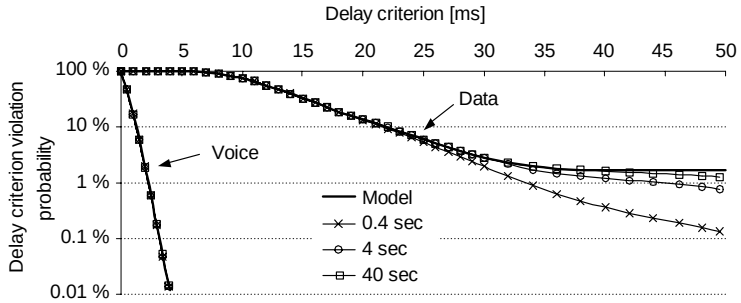

Fig. 4. Simulation of voice and data packet delays. The simulation of Markov modulated sources (see curves named "0.4 sec", "4 sec" and "40 sec") takes into account the correlation structure of the ON-OFF traffic. The model (see curve named "Model") does not take into account this.

To demonstrate the behavior of the proposed model, we use another simulator which enables us to take into account the above mentioned correlation structure. We simulated different ON-OFF sources. Figure 4 compares delays of Markov modulated sources with average ON period lengths of 0.4 sec, 4 sec and 40 sec to the result with the proposed model. In the simulations, the following two traffic classes are considered: high priority *voice* ($TTI_1 = 20$ ms, $b_1 = 40$ bytes, $\alpha_1 = 0.6$) and low priority *data* ($TTI_2 = 40$ ms, $b_2 = 360$ bytes, $\alpha_2 = 1$). In each simulation, 30 voice and 2 data sources were multiplexed. The buffer sizes were 7800 bit each, the server capacity was $C = 520$ kbps and the segment

size was $s = 100$ bytes. Up to $\tilde{D} \approx 30$ ms the delays hardly depend on the mean length of ON periods, and delay violations are dominated by the superimposed packet arrivals. For larger $\tilde{D}$ values the proposed model is conservative, and the delay violation is mainly caused by filling up the buffer from too many active sources. This example indicates that the model is conservative, and it is also accurate for strict delay criteria.

III. CAC ALGORITHM

In this section a CAC algorithm is proposed for the previously defined queueing model. The task of the CAC algorithm is to check on-line whether a certain traffic mix is within the admissible region[1]. In our case, when a new connection arrives, the CAC needs to check: the delay violation due to system overload ($\varepsilon_i^{overload}$), and the delay violation due to delayed packets ($\varepsilon_i^{delayed}$). For the sake of simplicity we divide the requirement $\tilde{\varepsilon}$ into two equal parts: $\tilde{\varepsilon}^{delayed} = \tilde{\varepsilon}^{overload} = 0.5 \cdot \tilde{\varepsilon}$. This way we need to ensure for each class that $\varepsilon_i^{delayed} \leq \tilde{\varepsilon}^{delayed}$ and that $\varepsilon_i^{overload} \leq \tilde{\varepsilon}^{overload}$. This is a conservative approach.

Delay violation due to system overload can be checked using (2). However, (2) is a complex algorithm which is difficult to evaluate in real-time. Fast approximations of (2) are described e.g. in [16]. In the numerical examples of the present paper, (2) is used to check $\varepsilon_i^{overload}$.

In this paper we focus on checking delay violation due to delayed packets. A number of articles dealing with periodic traffic sources propose to approximate the delay-limited borders of the admissible region by hyper-planes. The authors of [25] developed admission control methods for an ATM switch based on their observation that the linear approximation of the admissible region is acceptable. They verified this observation through numerical investigations. Kelly [28] approximated the arrival process of superposition of homogeneous periodic traffic sources by a Brownian bridge process, and showed that the border of the delay-limited admissible region is linear. This analysis has been extended in [16] for heterogeneous periodic ON-OFF sources, where the packet size, the period length, and the activity factor can be different. The above papers considered only FIFO scheduling. We have studied the applicability of the hyper-plane characterization for priority scheduling in UTRAN. Based on extensive simulations of the proposed model, we concluded that for priority scheduling the borders of the delay-limited admissible region can also be efficiently described by hyper-planes.

[1]The admissible region is the set of connection mixes $\{N_1, N_2, \ldots, N_K\}$ that can be served such that the QoS requirements are met.

Using these results we approximate the delay limited border by the intersection of hyper-planes[2]. One hyper-plane is associated with the delay requirement of each traffic class. Since we have K traffic classes and thus a K-dimensional space, K points span a hyper-plane. As we will see in Section IV, calculations of these K points of the j^{th} hyper-plane are simple if we choose these K points with the following co-ordinates:

$$\begin{aligned} \Theta_{ij} \quad &: \quad (\ x_1 = 0,\ \ldots\ ,\ x_i = \Lambda_{ij},\ \ldots\ ,\ x_j = 1,\ \ldots\ ,\ x_K = 0\ ), \quad i \neq j, \\ \Theta_{jj} \quad &: \quad (\ x_1 = 0,\ \ldots\ ,\ x_j = \Lambda_{jj} + 1,\ \ldots\ ,\ x_K = 0\ ), \end{aligned} \tag{4}$$

where Λ_{ij} is the x_i co-ordinate of the j^{th} hyper-plane, assuming that $x_j = 1$ and all other co-ordinates are zeroes. Note that the Λ_{ij} values are not necessarily integers, and that x_j co-ordinate of the last point is $\Lambda_{jj} + 1$ to avoid unnecessary calculation complexity.

In order to be able to calculate these co-ordinates we introduce N_{ij}^{max}. Let N_{ij}^{max} be the maximum number of connections from class i, assuming that the QoS requirement of one additional connection from class j is fulfilled, and there is no connection from other classes:

$$\begin{aligned} N_{ij}^{max} \quad &\triangleq \quad \max\left\{N_i :\ \varepsilon_j^{delayed} \leq \tilde{\varepsilon}^{delayed},\ N_j = 1, N_k = 0, k \notin \{i,j\}, i \neq j\right\}, \\ N_{jj}^{max} \quad &\triangleq \quad \max\left\{N_j :\ \varepsilon_j^{delayed} \leq \tilde{\varepsilon}^{delayed},\ N_k = 0, k \neq j\right\} - 1. \end{aligned} \tag{5}$$

As the utilization of a priority system tends to 100%, the probability that a low-priority packet will never be served also tends to 1. However, if classes i and j are on the same priority level, packet delays can still be low at 100% utilization, because the periodic packet arrivals can result in a rather limited burstiness of the aggregate traffic. This way, the point Θ_{ij} can also be outside of the overload-limited admissible region. In this case, the latter follows from that we use the calculation methods for evaluating the probability of delay criterion violation, $\varepsilon_j^{delayed}$, also in the overloaded system. In the overloaded system, the stationary delay distribution does not exist, but the results of the formulae can be interpreted as the complementary distribution of the delay of a class j packet that arrives into a *transient* system at $t = TTI_i$, which is empty at $t = 0$, and N_i uniformly distributed class i packets arrive into it in the interval $[0, TTI_i)$ [16], [19]. On the other hand, if classes i and j are on different priority levels, point Θ_{ij} is always inside the overload-limited admissible region (the region constrained only by (2)). In this case, according to the definitions of Λ_{ij} and N_{ij}^{max}, the Λ_{ij} values are bounded as $N_{ij}^{max} \leq \Lambda_{ij} < N_{ij}^{max} + 1$.

[2]The hyper-planes are represented in a co-ordinate system, where the axes mean the number of connections from K different traffic classes. The hyper-planes may cut the axes at non-integer values.

Input	
$\underline{N}$,	the number of connections
$\underline{\alpha}$, $\underline{b}$, $\underline{TTI}$, $\underline{p}$, $\underline{\tilde{D}}$,	traffic descriptors
C, s, $\tilde{\varepsilon}^{overload}$, $\tilde{\varepsilon}^{delayed}$,	system parameters
Output	
Decision,	admission decision (Accept or Reject)

Decision = **CACoverload**$(\underline{N}, \underline{\alpha}, \underline{b}, \underline{TTI}, \underline{p}, C, \tilde{\varepsilon}^{overload})$
If Decision = Accept **Then**
  For $j = 1$ **To** K
    If $N_j > 0$ **Then**
      Sum $= 0$
      For $i = 1$ **To** K
        $\Lambda_{ij} = \mathbf{CalcLambda}(i, j, \underline{\alpha}, \underline{b}, \underline{TTI}, \underline{p}, \underline{\tilde{D}}, C, s, \tilde{\varepsilon}^{delayed})$
        Sum = Sum $+ N_i/\Lambda_{ij}$
      End For
      If Sum $\cdot \Lambda_{jj} > \Lambda_{jj} + 1$ **Then** Decision = Reject
    End If
  End For
End If

TABLE I

THE FORMAL DESCRIPTION OF THE CAC ALGORITHM

Applying the above hyper-plane approximation, the necessary condition of accepting the traffic mix $(N_1, N_2, \ldots, N_K)$ is that for all j where $N_j > 0$, the following inequality must be met:

$$\sum_{i=1}^{K} \frac{\Lambda_{jj}}{\Lambda_{ij}} \cdot N_i \leq \Lambda_{jj} + 1. \tag{6}$$

The formal description of the CAC algorithm is given in Table I. Algorithms for checking the delay violation due to system overload (**CACoverload**) can be found in [16]. The formulae used for **CalcLambda** can be found in Section IV, and are summarized in (7):

$$\Lambda_{ij} \approx \begin{cases} (15) & \text{if class } i \text{ and class } j \text{ are on the same priority level,} \\ (25) & \text{if class } i \text{ has higher priority than class } j, \\ \infty & \text{otherwise.} \end{cases} \tag{7}$$

IV. CALCULATION ALTERNATIVES, CLOSED-FORM FORMULAE

In this section closed-form formulae are presented for calculating Λ_{ij}. We differentiate three cases depending on the priority levels of class i and class j.

A. Class i and class j are on the same priority level

If class i and class j are on the same priority level, then their packets share a common FIFO buffer. For approximating the Λ_{ij} values, we need to calculate the delay of a class j packet. The delay of a class j packet is equal to the sum of the service time of the queue content (the workload V) at the arrival of the packet, and the service time of the packet, i.e. $D_j = V + b_j/C$. We assume that at the arrival of a class j packet there are only class i packets in the queue. It is not generally true, because class i may have a larger TTI period than class j, and then the buffer may be emptied only in every TTI_i period. Still, it is a good approximation, as shown in [16].

The complementary distribution function of the workload, $F(t, N_i, C)$, considering N_i class i connections and server capacity C, can be determined using a combinatorial approach, see e.g. [19]. If the activity factor of class i is 1, then:

$$F(t, N_i, C) = \sum_{t' < l \leq N_i} \binom{N_i}{l} \left(\frac{l - t'}{T}\right)^l \left(1 - \frac{l - t'}{T}\right)^{n_i - l} \frac{T - N_i + t'}{T - l + t'} \quad , \text{ if } \alpha_i = 1, \qquad (8)$$

where $T = C\ TTI_i/b_i$ and $t' = C\ t/b_i$.

For large N_i, the function $F(t, N_i, C)$ can be approximated by a direct formula, $F^{approx}(t, N_i, C)$, which is based on the periodic arrival process being approximated by a Brownian-bridge process (see e.g. [28]). If the activity factor of class i is 1, then:

$$F^{approx}(t, N_i, C) = \exp\left\{-\frac{2\ C\ t}{TTI_i\ N_i\ \rho_i^2}\left(\frac{C\ t}{TTI_i} + C - N_i\ \rho_i\right)\right\} \quad , \text{ if } \alpha_i = 1. \qquad (9)$$

If the activity factor of class i is less than 1, i.e. $\alpha_i < 1$, then the complementary distribution function of the workload is calculated as:

$$F_\alpha(t, N_i, C) = \sum_{n_i=0}^{N_i} \Pi_i(n_i) F(t, n_i, C), \qquad (10)$$

which is a tight approximation of (3).

We have the following three methods to obtain the Λ_{ij} value depending on the applied approximation in (10):

Method **A**:

In this method we use the function $F(t, n_i, C)$ and the exact calculation of $\Pi_i(n_i)$ in (10). Because both applied functions in (10) can be evaluated only for integers we apply a simple interpolation method for determining Λ_{ij}. Denote by λ_{ij}^* the largest possible number of connections that can be admitted, such that the delay requirement is met. It is the greatest integer solution of the inequality below:

$$\lambda_{ij}^* \;=\; \max\left\{\lambda_{ij} \;\middle|\; F_\alpha(\tilde{D}_j - b_j/C, \lambda_{ij}, C) \le \tilde{\varepsilon}^{delayed}\right\}. \tag{11}$$

We know that $\lambda_{ij}^* \le \Lambda_{ij} \le \lambda_{ij}^* + 1$. Therefore, if λ_{ij}^* connections were in the system, the capacity C could be still decreased (to C_1) while the delay requirements were fulfilled. Similarly, if $\lambda_{ij}^* + 1$ connections were in the system, the capacity C should be increased (to C_2) in order to fulfill the delay requirements. Using C, C_1 and C_2 ($C_1 \le C \le C_2$) we approximate Λ_{ij} as:

$$\Lambda_{ij} \approx \lambda_{ij}^* + \frac{C - C_1}{C_2 - C_1}, \tag{12}$$

where C_1 and C_2 are the solutions of the following equations in c_1 and c_2, respectively:

$$C_1 \;=\; c_1 \quad : \quad F_\alpha(\tilde{D}_j - b_j/c_1, \lambda_{ij}^*, c_1) = \tilde{\varepsilon}^{delayed}, \tag{13}$$

$$C_2 \;=\; c_2 \quad : \quad F_\alpha(\tilde{D}_j - b_j/c_2, \lambda_{ij}^* + 1, c_2) = \tilde{\varepsilon}^{delayed}. \tag{14}$$

Method **B**:

This method is similar to method **A**, but here the function $F^{approx}(t, n_i, C)$ is used in (10) instead of function $F(t, n_i, C)$.

Method **C**:

In this method we use the $F^{approx}(t, n_i, C)$ and the Gaussian approximation of $\Pi_i(n_i)$ in (10). This way, we obtain a closed-form formula for Λ_{ij} (for details see [16]):

$$\Lambda_{ij} \approx \frac{C\ TTI_i + C\ \alpha_i\ \left(\tilde{D}_j - b_j/C\right)}{\alpha_i\ b_i} \cdot \left(1 - \frac{b_i\ ln(\tilde{\varepsilon}^{delayed})}{2\ C\ \left(\tilde{D}_j - b_j/C\right)}\right)^{-1}. \tag{15}$$

B. Class i has higher priority than class j

In this section we look for methods **A**, **B** and **C** in the priority system, where class i has higher priority than class j. In this case, both classes have a separate buffer. The delay of a class j packet is the sum of the queueing delay Q_j and the service time of its last segment: $D_j = Q_j + s_j^{last}/C$, where s_j^{last} is the size of the last segment of the class j packet[3]. We

[3] $s_j^{last} = b_j - \lfloor (b_j - 1)/s \rfloor \cdot s$

assume that the class j packet arrives to the lower priority queue at time $t = 0$. The queueing delay is as follows:

$$Q_j = \inf\{q \geq 0 : \quad qC \geq b_j - s_j^{last} + A_i(q) + V_i(0) - V_i(q)\}, \tag{16}$$

where $A_i(t)$ is the amount of traffic arriving to the high priority queue in the interval $[0, t)$ and $V_i(t)$ is the workload in the higher priority queue at time t. The probability density function of the queueing delay is as follows:

$$f_Q(t) = \lim_{\Delta t \to 0} \frac{Pr\left\{V_i(t) = 0,\ b_j - s_j^{last} - C\Delta t \leq \mathbf{Idle}(t) < b_j - s_j^{last}\right\}}{\Delta t}, \tag{17}$$

because the service of the last segment of the class j packet can only start at time t if the availability time (or idle time) of the server in the $(0, t)$ interval seen by the class j packet, denoted as $\mathbf{Idle}(t)$, just falls short of being able to serve $b_j - s_j^{last}$ bits, but at the time t the higher priority queue is empty. This density function has been evaluated in [23]. However, the obtained formula is rather complex, and can only be calculated efficiently for small systems. Since the admission control algorithm also has to be applied for large systems, we have to rely on approximation.

The complementary distribution function of the queueing delay, $G(t, N_i, C)$, considering N_i class i connections and server capacity C, can be approximated by the empty buffer approximation.

First, we assume that the activity factor of class i is 1. Considering (16), the arrival process, $A_i(t)$, is a known process. Due to periodicity of packet arrivals, the $V_i(t)$ is periodic and depends on the arrival process in the interval $[0, TTI_i]$. The main difficulty when evaluating $G(t, N_i, C)$ is caused by this dependency. In [27], the *empty buffer (EB)* approximation has been used to obtain workload bounds for priority systems. When using the EB approximation we consider what the class j delay would be if there was not any accumulation of class i workload (i.e., $V_i(t) = 0$ for all t), as would occur with constant rate fluid input with stable higher priority queue (i.e., $N_i\rho_i < C$). Using the EB approximation the complementary distribution function of the queueing delay is approximated by:

$$G^{approx}(t, N_i, C) = 1 - \Pr\{A_i(t) + b_j - s_j^{last} \leq tC\} \quad , \text{ if } \alpha_i = 1. \tag{18}$$

The event that the last segment of the class j packet could not be served before time $\tilde{D}_j$ is equivalent to the event that all segments before the last one could not be served before $\tilde{Q}_j = \tilde{D}_j - s_j^{last}/C$. As in Section IV-A, we use the Brownian bridge approximation of the

arrival process $A_i(t)$ in order to obtain a simple formula for calculating Λ_{ij}. Doing this, the following result is obtained:

$$G^{approx}(t, N_i, C) = \Phi\left\{b_j - s_j^{last};\ (C - N_i\rho_i)t,\ N_i\rho_i^2 t(TTI_i - t)\right\} \quad , \text{ if } \alpha_i = 1, \tag{19}$$

where $t \leq TTI_i$ and $\Phi\{t; \mu, \sigma^2\}$ is the normal distribution with mean μ and variance σ^2.

Secondly, we assume that the activity factor of the N_i higher priority connections is smaller than one ($\alpha_i < 1$). As in Section IV-A, we extend the method to handle class i connections, which have activity factor smaller than 1:

$$G_\alpha^{approx}(t, N_i, C) = \sum_{n_i=0}^{N_i} \Pi_i(n_i) G^{approx}(t, n_i, C). \tag{20}$$

We have the following three methods to obtain the Λ_{ij} value depending on the applied approximation in (20):

Method **A**:

In this method we use simulation to determine the Λ_{ij} value.

Method **B**:

In this method we use (20). Because the function $\Pi_i(n_i)$ in (20) can be evaluated only for integers, we apply the following interpolation method (similar to (12)) for determining Λ_{ij}:

$$\Lambda_{ij} \approx \lambda_{ij}^* + \frac{C - C_1}{C_2 - C_1}, \tag{21}$$

where λ_{ij}^* is the greatest integer solution of the inequality below:

$$\lambda_{ij}^* \ = \ \max\left\{\lambda_{ij} \ \middle|\ G_\alpha^{approx}(\tilde{D}_j - s_{last}/C, \lambda_{ij}, C) \leq \tilde{\varepsilon}^{delayed}\right\}, \tag{22}$$

and C_1 and C_2 are the solutions of the following equations in c_1 and c_2, respectively:

$$C_1 \ = \ c_1 \quad : \quad G_\alpha^{approx}(\tilde{D}_j - s_{last}/c_1, \lambda_{ij}^*, c_1) = \tilde{\varepsilon}^{delayed}, \tag{23}$$

$$C_2 \ = \ c_2 \quad : \quad G_\alpha^{approx}(\tilde{D}_j - s_{last}/c_2, \lambda_{ij}^* + 1, c_2) = \tilde{\varepsilon}^{delayed}. \tag{24}$$

Method **C**:

In this method we use the Gaussian approximation of $\Pi_i(n_i)$ in (20). This way, we obtain a closed-form approximation for Λ_{ij}:

$$\Lambda_{ij} \approx \frac{g_{ij} - h_j - \sqrt{g_{ij}(g_{ij} - 2h_j)}}{\alpha_i \rho_i \tilde{Q}_j}, \tag{25}$$

where

$$g_{ij} = \left(\mathbf{Erfc}^{-1}(2\tilde{\varepsilon}^{delayed})\right)^2 \rho_i\Big(TTI_i - \alpha_i\tilde{Q}_j\Big), \quad h_j = b_j - s_j^{last} - C\tilde{Q}_j.$$

C. Class i has lower priority

A precise solution of this case is presented in [29]. To avoid serious performance implications of low priority segments being under service on high priority segments, the segment size is proposed to be around 50-100 bytes on low-rate links. Since the segment size is small, we neglect the effect of a segment possibly under service from a low priority packet on the delay of a high priority packet. It means that Λ_{ij} values for this case are set to infinity (i.e. $\Lambda_{ij} = \infty$ in (6)).

V. Numerical examples

In this section, numerical examples are provided which demonstrate the applicability of the proposed CAC algorithm, and study the accuracy of the CAC and the consequences of using priority scheduling.

A. The use of Λ_{ij} values

To demonstrate the meaning of Λ_{ij} values, we present admissible regions for two classes. The traffic classes are the following:

- class i is *voice*: $TTI_i = 20$ ms, $b_i = 336$ bit, $\alpha_i = 0.55$, $\tilde{D}_i = 5$ ms and
- class j is *64 kbps data*: $TTI_j = 20$ ms, $b_j = 1480$ bit, $\alpha_j = 1$, $\tilde{D}_j = 7.5$ ms.

Further parameters are: $\tilde{\varepsilon}^{overload} = \tilde{\varepsilon}^{delayed} = 0.0005$, $C = 1504$ kbps, $s = 800$ bits.

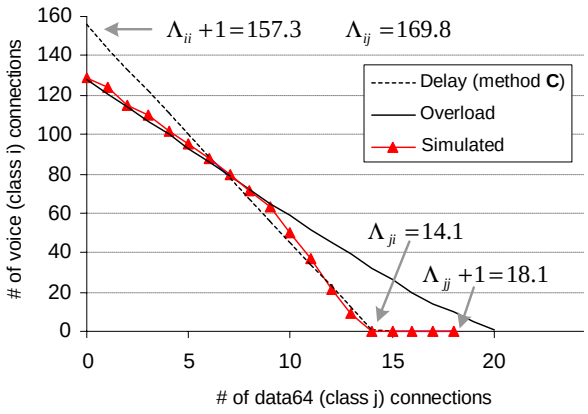

Fig. 5. Example: admissible region with FIFO scheduling

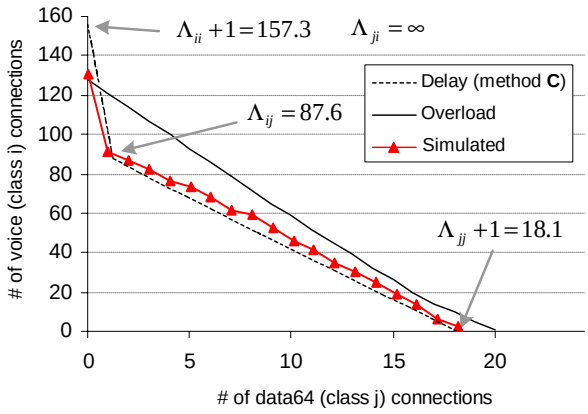

Fig. 6. Example: admissible region with priority scheduling

The Λ_{ij} values in Figure 5 and Figure 6 have been calculated using method **C**. In the case of Figure 5, packets of both classes are sent to the same buffer and FIFO scheduling

is applied. In the case of Figure 6, priority scheduling is used and voice has higher priority. The admissible region which is obtained by simulation (in the figures it is identified by "Simulated") is well approximated by the intersection of delay-limited ("Delay") and overload-limited ("Overload") admissible regions. Since the difference between the target delays $\tilde{D}_i$ and $\tilde{D}_j$ is not large enough, the advantages of priority scheduling can not be exploited. Examples, where priority scheduling performs better than FIFO are provided in Section V-C.

B. The accuracy of the CAC algorithm

The CAC algorithm is accurate if it guarantees QoS requirements for the admitted connections and at the same time admits nearly as many connections as possible.

Consider the traffic classes defined in Table II and the following system parameters: $C = 1504$ kbps, $s = 800$ bit, $\tilde{\varepsilon}^{overload} = \tilde{\varepsilon}^{delayed} = 0.0005$. The target delay of high priority RBs (voice and DCCH) is $\tilde{D}_{high} = 5$ ms and of low priority RBs (64k RB, 384k RB, PCH, FACH1 and FACH2) is $\tilde{D}_{low} = 7.5$ ms. Considering signaling traffic (PCH, FACH and DCCH), in all numerical examples there are 3 PCH, 3 FACH1 and 3 FACH2 connections in the system, while the number of DCCH connections equals the sum of the number of voice, 64k RB and 384k RB connections (on roles of signaling channels, see [11]).

TABLE II

PARAMETERS OF THE NUMERICAL EXAMPLE

RB type	TTI [ms]	b [bit]	α	priority level
voice	20	336	0.55	high
64k RB	20	1480	1	low
384k RB	10	4360	1	low
DCCH	40	216	0.2	high
PCH	10	480	0.5	low
FACH1	10	432	0.5	low
FACH2	10	456	0.5	low

Admissible regions determined using different methods are compared with simulation and also with a complex analytical method ([19], pages 409-411). The latter method will be referred to as "COST". The results are presented in Table III and Figure 7. FIFO scheduling is used only, because the COST method can not work with priorities. In the first column,

the results achieved by simulation are shown. In the second column, the results achieved by the COST method are given. In the last three columns of Table III the results calculated by the proposed CAC algorithm using methods **A**, **B** and **C** are presented. The COST method is an approximation based on the general Beneš result and a large deviation approximation, but it cannot be calculated in real-time. The good performance of this method would make it suitable for applying it in a CAC, however its complexity grows rapidly with the number of connections and traffic classes. Therefore, practically it can not be used in a CAC.

TABLE III

NUMBER OF ADMITTED VOICE CONNECTIONS IN THE FIFO SYSTEM USING DIFFERENT METHODS, AND BETWEEN BRACKETS, THE LINK UTILIZATION IN PERCENTAGE OF THE LINK CAPACITY ("–" MEANS THAT THE TRAFFIC MIX CAN NOT BE ACCEPTED, "0" MEANS THAT THE TRAFFIC MIX CAN BE ACCEPTED AND THE NUMBER OF VOICE CONNECTIONS IN THE MIX IS 0.)

	0 of 384k RB	**1** of 384k RB
0 of 64k RB	92(77), 90(75), 90(75), 90(75), 90(75)	54(80), 53(79), 35(67), 35(67), 30(63)
1 of 64k RB	84(76), 84(76), 84(76), 84(76), 84(76)	47(80), 46(79), 24(64), 25(65), 20(61)
2 of 64k RB	78(77), 77(76), 77(76), 77(76), 77(76)	41(81), 37(78), 14(62), 15(63), 10(60)
3 of 64k RB	72(78), 70(77), 71(77), 71(77), 71(77)	34(81), 19(71), 3(60), 5(61), 0(58)
4 of 64k RB	66(79), 64(78), 64(78), 64(78), 64(78)	23(78), 9(69), – , – , –
5 of 64k RB	59(79), 56(77), 58(78), 58(78), 58(78)	3(70), 0(68), – , – , –
6 of 64k RB	53(80), 43(73), 52(79), 52(79), 52(79)	0(73), – , – , – , –
7 of 64k RB	45(79), 35(73), 44(79), 45(79), 45(79)	– , – , – , – , –
8 of 64k RB	39(80), 31(75), 33(76), 39(80), 39(80)	– , – , – , – , –
9 of 64k RB	33(81), 25(76), 22(74), 30(79), 31(80)	– , – , – , – , –
10 of 64k RB	21(78), 20(77), 11(71), 20(77), 21(78)	– , – , – , – , –
11 of 64k RB	11(76), 11(76), 0(69), 10(75), 11(76)	– , – , – , – , –
12 of 64k RB	0(74), 0(74), – , 0(74), 1(74)	Sim, COST, **A**, **B**, **C**

The accuracy of the proposed CAC method gets better as the size of the system[4] increases. This is, because in methods **A** and **B**, the possible error of the interpolation (such as (12)) vanishes for large Λ_{ij} values. In addition, the Gaussian approximations applied in methods **B** and **C** are also more accurate having larger number of connections. Figure 7 demonstrates these observations: If there are no 384k RBs in the system, the CAC performs

[4]The size of the system from the point of view of a connection mix is measured by the number of admittable connections from the connection type with the largest peak-rate in that mix

well irrespectively of the used calculation alternative. But since only a single 384k RB can be admitted, this system is small from the point of view of this large bearer. Therefore, if there is one 384k RB in the system, the errors of interpolation and the Gaussian approximation result in lower accuracy.

TABLE IV

NUMBER OF ADMITTED VOICE CONNECTIONS IN THE PRIORITY SYSTEM USING DIFFERENT METHODS, AND BETWEEN BRACKETS, THE LINK UTILIZATION IN PERCENTAGE OF THE LINK CAPACITY

	0 of 384k RB	**1** of 384k RB	**2** of 384k RB
0 of 64k RB	88(74), 81(69), 76(66), 78(67)	47(75), 38(69), 40(70), 41(71)	16(83), 10(79), 8(77), 3(74)
1 of 64k RB	79(73), 69(66), 67(65), 69(66)	44(78), 35(72), 33(70), 31(69)	10(84), 3(79), 0(77), –
2 of 64k RB	72(73), 64(68), 62(66), 64(68)	39(79), 31(74), 29(73), 27(71)	4(84), – , – , –
3 of 64k RB	68(75), 60(70), 58(68), 59(69)	34(81), 26(76), 24(74), 22(73)	– , – , – , –
4 of 64k RB	62(76), 55(71), 53(70), 55(71)	29(83), 22(78), 20(76), 17(74)	– , – , – , –
5 of 64k RB	57(78), 50(73), 49(72), 50(73)	22(83), 17(79), 15(78), 13(77)	– , – , – , –
6 of 64k RB	52(79), 45(74), 44(74), 45(74)	16(84), 12(81), 10(80), 8(78)	– , – , – , –
7 of 64k RB	45(79), 40(76), 39(75), 41(77)	10(85), 7(82), 6(82), 3(80)	– , – , – , –
8 of 64k RB	39(80), 36(78), 35(78), 36(78)	4(85), 2(84), 1(83), –	– , – , – , –
9 of 64k RB	34(82), 31(80), 30(79), 31(80)	– , – , – , –	– , – , – , –
10 of 64k RB	27(82), 26(81), 26(81), 26(81)	– , – , – , –	– , – , – , –
11 of 64k RB	21(83), 20(82), 20(82), 20(82)	– , – , – , –	– , – , – , –
12 of 64k RB	14(83), 14(83), 14(83), 14(83)	– , – , – , –	– , – , – , –
13 of 64k RB	8(84), 8(84), 8(84), 8(84)	– , – , – , –	– , – , – , –
14 of 64k RB	2(85), 2(85), 2(85), 2(85)	– , –, – , –	Sim , **A , B , C**

The conservativeness resulting from dividing the target QoS violation probability, $\tilde{\varepsilon} = 0.001$, into two parts ($\tilde{\varepsilon}^{delayed}$ and $\tilde{\varepsilon}^{overload}$) can be easily followed if there are no 64k and 384k RBs in the system: instead of the possible 92 connections, only 90 are accepted by the CAC.

Table IV and Figure 8 present the admissible region if priority scheduling is used. The first admissible region (corresponding to the first column in the table) is determined by simulation, while the others are determined using the proposed CAC algorithm with different calculation methods. Considering method **A**, Λ_{ij} values with priority levels $p_i < p_j$ are obtained by simulation. The CAC with method **C** results in a somewhat more conservative admissible region when the traffic mix is dominated by a few large bearers (for example, 384k RB connections).

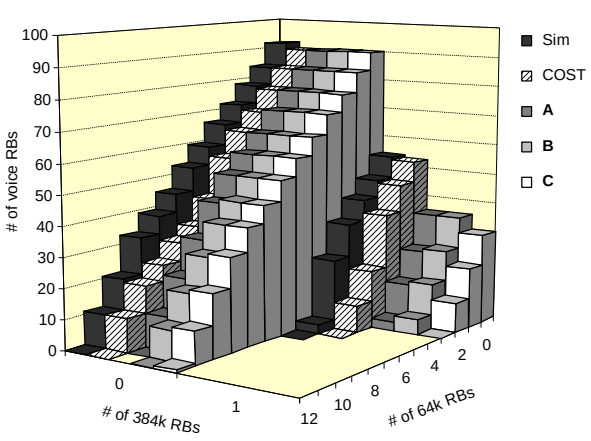

Fig. 7. Number of admitted voice connections in the FIFO system using different algorithms (the graphical representation of Table III)

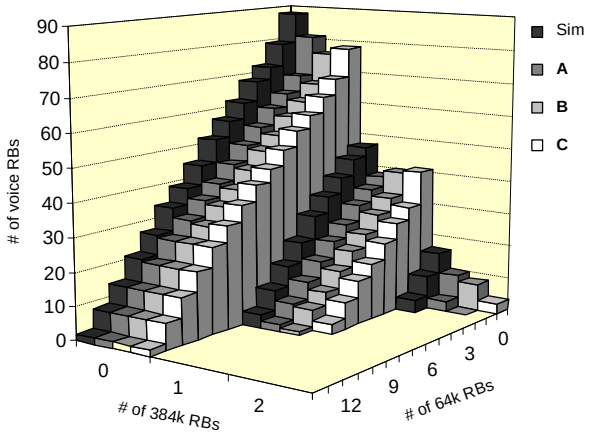

Fig. 8. Number of admitted voice connections in the priority system using different algorithms (the graphical representation of Table IV)

We investigated numerous examples using UTRAN specific traffic classes and typical server rates. We have found that the CAC guarantees QoS while admitting nearly as many connections as possible. Connections, which could be served by the system but still got rejected due to the inaccuracy of the CAC, are rare, because in typical states[5] of the system the CAC is accurate.

C. The effect of priority scheduling

In this section, we demonstrate by an example that it can be worth using priority scheduling in UTRAN. The following FIFO and priority systems are examined:

- $FIFO$: FIFO scheduling, $\tilde{D}_{high} = 5$ ms and $\tilde{D}_{low} = 7.5$ ms,
- $Prio^{7.5ms}$: priority scheduling, $\tilde{D}_{high} = 5$ ms and $\tilde{D}_{low} = 7.5$ ms,
- $Prio^{10ms}$: priority scheduling, $\tilde{D}_{high} = 5$ ms and $\tilde{D}_{low} = 10$ ms,
- $Prio^{15ms}$: priority scheduling, $\tilde{D}_{high} = 5$ ms and $\tilde{D}_{low} = 15$ ms.

The system parameters and the traffic classes are the same as in the previous section. Table V and Figure 9 present admissible regions obtained using method **C**.

It is clear from the results that prioritization (QoS separation) increases the number of admissible low-priority connections as $\tilde{D}_{low}$ gets larger. If $\tilde{D}_{low}$ is close to $\tilde{D}_{high}$, then the number of admissible high-priority connections may decrease even if there are no 64k RB

[5]It is not likely that many large-bitrate connections dominate in the system.

TABLE V

NUMBER OF ADMITTED VOICE CONNECTIONS IN CASES $Prio^{15ms}$, $Prio^{10ms}$, $Prio^{7.5ms}$ AND $FIFO$

	0 of 384k RB	**1** of 384k RB	**2** of 384k RB
0 of 64k RB	90, 86, 78, 90	53, 53, 41, 30	16, 16, 3, –
1 of 64k RB	84, 79, 69, 84	46, 46, 31, 20	10, 10, –, –
2 of 64k RB	77, 74, 64, 77	40, 40, 27, 10	4, 4, –, –
3 of 64k RB	71, 70, 59, 71	34, 34, 22, 0	–, –, –, –
4 of 64k RB	64, 64, 55, 64	27, 27, 17, –	–, –, –, –
5 of 64k RB	58, 58, 50, 58	21, 21, 13, –	–, –, –, –
6 of 64k RB	52, 52, 45, 52	15, 15, 8, –	–, –, –, –
7 of 64k RB	45, 45, 41, 45	9. 9, 3, –	–, –, –, –
8 of 64k RB	39, 39, 36, 39	3, 3, –, –	–, –, –, –
9 of 64k RB	33, 33, 31, 31	–, –, –, –	–, –, –, –
10 of 64k RB	26, 26, 26, 21	–, –, –, –	–, –, –, –
11 of 64k RB	20, 20, 20, 11	–, –, –, –	–, –, –, –
12 of 64k RB	14, 14, 14, 1	–, –, –, –	–, –, –, –
13 of 64k RB	8, 8, 8, –	–, –, –, –	–, –, –, –
14 of 64k RB	2, 2, 2, –	–, –, –, –	$Prio^{15}$, $Prio^{10}$, $Prio^{7.5}$, FIFO

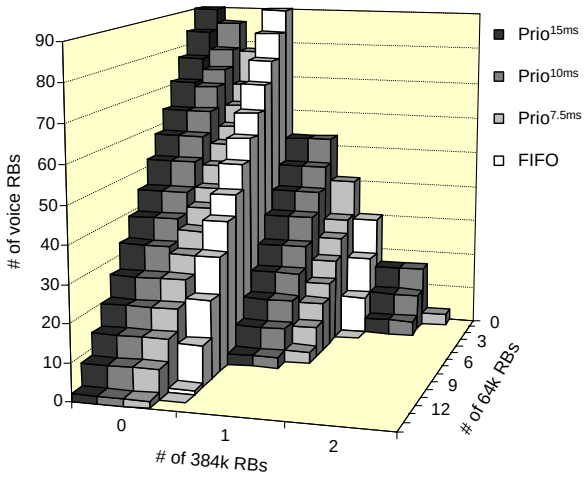

Fig. 9. Number of admitted voice connections in cases $Prio^{15ms}$, $Prio^{10ms}$, $Prio^{7.5ms}$ and $FIFO$ (the graphical representation of Table V)

and 384k RB connections in the system, because the low priority signaling traffic (e.g. PCH) is always present. From this numerical example, we can conclude that priority scheduling is more advantageous than FIFO, already with $\tilde{D}_{low} \geq 10$ ms. In the case of the $Prio^{15ms}$, we found that only the system overload determined the admissible region.

VI. CONCLUSION

In this paper a connection admission control algorithm for UTRAN is proposed. The CAC algorithm works for priority scheduling.

The proposed CAC method is accurate, and guarantees QoS. It works in real-time due to the developed novel closed-form formulae. The validation of the proposed CAC is carried out by simulation. We also provided a performance evaluation study with numerical examples for comparing the accuracy of different calculation alternatives within the CAC, and discussed briefly the effect of priority scheduling.

Based on the results we suggest the practical application of our method.

REFERENCES

[1] ITU, AAL Type 2 Signalling Protocol (Capability Set 3), Q.2630.3, October, 2003

[2] 3GPP, IP Transport in UTRAN, TR 25.933 V5.2.0, September, 2002

[3] G. Eneroth, et al., Applying ATM/AAL2 as a Switching Technology in 3rd Generation Mobile Networks, *IEEE Communication Magazine*, 37(6):112-122, 1999

[4] 3GPP, Quality of Service (QoS) concept and architecture, TS 23.107 V3.9.0, September, 2002

[5] 3GPP, Synchronization in UTRAN (Stage 2), TS 25.402 V5.1.0, June, 2002

[6] 3GPP, Common test environments for User Equipment (UE), TS 34.108 V4.7.0, June, 2003

[7] 3GPP, UTRAN Iub/Iur interface user plane protocol for DCH data streams, TS 25.427 V5.1.0, December, 2002

[8] 3GPP, UTRAN Iub interface user plane protocols for Common Transport Channel data streams, TS 25.435 V5.5.0, June, 2003

[9] 3GPP, RRC protocol specification, TS 25.331 V5.5.0, June, 2003

[10] 3GPP, Delay budget within the Access Stratum, TR 25.853 V4.0.0, March, 2001

[11] 3GPP, Multiplexing and channel coding (FDD), TS 25.212 V5.0.0, March, 2002

[12] E. Dahlman et. al., WCDMA – The Radio Interface for Future Mobile Multimedia Communications, *IEEE Transactions on Vehicular Technology*, vol. 47, No. 4, pp. 1105-1118, Nov. 1998

[13] S. Anand and A. Chockalingam, Performance Analysis of Voice/Data Cellular CDMA with SIR based Admission Control, *IEEE Journal on Selected Areas in Communications*, vol. 21, No. 10, Dec. 2003

[14] J. Prez-Romero, et al., A Downlink Admission Control Algorithm for UTRA-FDD, *4th IEEE Conference on Mobile and Wireless Communications Networks*, Stockholm, Sweden, 2002

[15] J. Lee and Y. Han, Downlink Admission Control for Multimedia Services in WCDMA, *PIMRC 2002*, Lisbon, Portugal, 2002

[16] Sz. Malomsoky, S. Rácz and Sz. Nádas, Connection Admission Control in UMTS Radio Access Networks, *Computer Communications - Special Issue: 3G Wireless and Beyond*, pp. 2011-2023, Vol. 26, November, 2003

[17] R. Makké, et al., Performance of the AAL2 Protocol within the UTRAN, *IEEE ECUMN*, 2002

[18] A-F. Canton, et al., Performance Analysis of AAL2/ATM in UMTS Radio Access Network, *IEEE PIMRC*, 2002

[19] J. Roberts, U. Mocci, and J. Virtamo, eds., Broadband network teletraffic, *Final report of action COST 242*, Springer Verlag, 1996

[20] I. Norros, J. W. Roberts, A. Simonian, and J. T. Virtamo, The Superposition of Variable Bit Rate Sources in an ATM Multiplexer, *IEEE JSAC*, Vol. 9., No. 3., 378-387, 1991

[21] A-F. Canton, et al., Statistical Analysis of Weighted Round Robin Service Differentiation at AAL2 Layer in UMTS Radio Access Network, *IEEE Globecom*, 2002

[22] M. Menth, et al., MEDF - A Simple Scheduling Algorithm for Two Real-Time Transport Service Classes with Application in the UTRAN *IEEE INFOCOM*, 2003

[23] G. Tóth and Cs. Antal, Segmentation and Packet Delays in Strict Priority Systems with CBR traffic, *QoS-IP 2003*, Milan, Italy, 2003

[24] G. Matéfi, J. Farkas and Cs. Antal, Towards Efficient Call Admission Control in IP UTRAN, *ITC18*, Berlin, Germany, 2003

[25] Linhai He and Albert Wong, Connection Admission Control Design for GlobeView-2000 ATM Core Switches, *Bell Labs Technical Journal*, pp.94-110, January-March, 1998

[26] A. W. Berger and W. Whitt, Effective Bandwidths with Priorities, *IEEE/ACM Transactions on Networking*, vol. 6, No. 4, pp. 447-460, August 1998

[27] A. W. Berger and W. Whitt, Workload Bounds in Fluid Models with Priorities, *Performance Evaluation*, vol. 41, pp. 249-267, 2000

[28] F. P. Kelly, Notes on Effective Bandwidths, *Stochastic Networks: Theory and Applications*, Vol. 4. of Royal Stat. Soc. Lecture Notes Series, Oxford University Press, 141-168, 1996

[29] K. Iida, et al., Delay Analysis for CBR Traffic under Static-Priority Scheduling, *IEEE/ACM Transactions on Networking*, 9(2), April, 2001