The final publication is available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6092416. DOI 10.1109/TMC.2011.256 © 2011 IEEE

On the Real-Time Hardware Implementation Feasibility of Joint Radio Resource Management Policies for Heterogeneous Wireless Networks

M.C. Lucas-Estañ and J. Gozalvez

Abstract— The study and design of Joint Radio Resource Management (JRRM) techniques is a key and challenging aspect in future heterogeneous wireless systems where different Radio Access Technologies will physically coexist. In these systems, the total available radio resources need to be used in a coordinated way to guarantee adequate satisfaction levels to all users, and maximize the system revenues. In addition to carry out an efficient use of the available radio resources, JRRM algorithms need to exhibit good computational performance to guarantee their future implementation viability. In this context, this paper proposes novel JRRM techniques based on linear programming techniques, and investigates their computational cost when implemented in DSP platforms commonly used in mobile base stations. The obtained results demonstrate the feasibility to implement the proposed JRRM algorithms in future heterogeneous wireless systems.

Index Terms— Heterogeneous wireless systems, Joint Radio Resource Management, DSP, embedded systems.

1 Introduction

THE evolution of mobile and wireless communication systems is being characterized by the coexistence of diverse Radio Access Technologies (RAT) with different, but sometimes complementary, technical characteristics. In parallel, novel user applications are continuously appearing with diverse Quality of Service (QoS) requirements. Despite the appearance of novel RATs with increasing performance, the research community agrees that future mobile and wireless communication systems will be composed of heterogeneous RATs physically coexisting and offering mobile services to a wide range of QoS-demanding users in a coordinated manner. In this context, a key aspect of future heterogeneous wireless systems is the coordinated management of heterogeneous radio resources, usually referred as Joint Radio Resource Management (JRRM) or Common Radio Resource Management (CRRM). The 3GPP (3rd Generation Partnership Project) defines the JRRM concept and describes different supporting network architectures that ensure the interoperability between the different access technologies ([1] and [2]). Novel JRRM policies need then to be designed so that the total available radio resources are efficiently distributed among active users in order to maximize the system revenue and provide the QoS levels demanded by users/services in multimedia environments. To carry out the most efficient use of the total available resources, JRRM policies must decide for each incoming call the RAT over which it will be conveyed (RAT selection) and the number of radio resources within the selected RAT (intra-RAT RRM) that will be necessary to satisfy the user/service QoS requests. Furthermore, the JRRM policy and resulting radio resource assignments should be capable to dynamically adapt to the current operating conditions, for example system load and active multimedia services.

Most of the JRRM studies reported in the literature focus on the design of initial RAT selection techniques. For example, [3] describes the framework over which JRRM algorithms can be developed, and proposes some basic techniques to address the initial RAT selection dilemma based on pre-established service-to-RAT assignments and user location. Other studies have investigated how to exploit multi-technology terminals capability to switch between RATs in order to free the capacity required to accept new calls from single-mode terminals. Several strategies to perform this traffic rearrangement are discussed in [4] and references therein. Another JRRM approach that has received much attention from the community is load balancing. For example, the JRRM load balancing mechanism reported in [5] aims at achieving a uniform traffic distribution between the available RATs. As the authors point out, such uniform distribution is desirable in order to maximize the trunking gain and minimize the probability of making unnecessary vertical handovers of multi-technology terminals between RATs. For non real-time services, the load balancing is performed based on the measured buffer delay, while the

• *M. C. Lucas-Estañ and J.Gozalvez are with the Ubiquitous Wireless Communications Research (Uwicore) Laboratory at the University Miguel Hernandez of Elche, Spain. E-mails: m.lucas@umh.es, j.gozalvez@umh.es.*

Manuscript received 17th January 2010.

Digital Object Indentifier 10.1109/TMC.2011.256

authors propose a mechanism based on load thresholds for real-time services. In [6], a RAT selection algorithm that assigns the user to the most suitable RAT is proposed. The RAT suitability is based on the current RAT load and a pre-established load threshold, which are empirically calculated with the target to achieve the maximum throughput gain. Finally, it is worthwhile highlighting a RAT selection proposal based on the species competition model [7]. The proposal adapts some RAT parameters (price and support bandwidth) according to their current operating conditions in order to attract or dismiss users from accessing them.

Proposals to jointly address the RAT selection and intra-RAT RRM dilemmas have also been reported. For example, [8] presents a Joint Call Admission Control (JCAC) algorithm that combines the RAT selection and Call Admission Control (CAC) mechanisms in order to reduce the call blocking and dropping probabilities, and ensure a fair radio resource allocation. Another interesting JRRM algorithm based on neural networks and fuzzy logic has been proposed in [9]. This algorithm simultaneously determines the most appropriate RAT and bit rate allocation considering factors such as signal strength, resource availability, and mobile speed. While [9] determines the necessary bit rate at the assigned RAT, it does not tackle the problem of intra-RAT radio resources allocation. In this context, the work reported in [10] and [11] proposed novel JRRM policies that simultaneously assign to each user an adequate combination of RAT and number of radio resources within such RAT to guarantee the user/service QoS requirements. The proposed JRRM techniques are based on linear programming and optimization techniques.

Previous studies have focused on evaluating the QoS performance that can be achieved by novel JRRM techniques, but have not investigated their computational cost and implementation feasibility. The evaluation of the computational efficiency of new proposals is widely conducted in other research fields like audio and video real-time compression, where the time spent by the algorithm to process the data is crucial to provide a good performance to the end user. Although time requirements are not as demanding as for audio and video compression techniques, JRRM decisions in mobile networks should be made as quickly as possible in order to be able to efficiently adapt the use of the radio resources to the current operating conditions. Given that JRRM decisions are based on an increasing number of variables and data, the JRRM processing time is becoming an important factor to be considered when designing novel and advanced JRRM algorithms. In the mobile communications field, the majority of studies are based on computer simulations, and few of them evaluate the hardware implementation and computational cost of novel techniques. As an example, Yavuz and Leung [12] measured the CPU time of their proposed admission control method running on a computer, and compared it with that obtained with previous admission control algorithms. To the authors' knowledge, there are currently no published studies of the hardware computational cost of JRRM policies, in particular of advanced JRRM policies jointly addressing the RAT selection and intra-RAT RRM dilemmas. In this context, this work presents the first hardware implementation of advanced JRRM policies to analyse their computational cost and the time they require to be executed on real hardware systems. By comparing their execution time to the time needed in current cellular networks to conduct a vertical handover or assign radio resources to new users, this study will provide valuable indications on the feasibility of implementing the proposed JRRM techniques in commercial networks. The result of this study is of relevance to the research community since it demonstrates the feasibility of implementing complex JRRM policies, and provides the first indications on their hardware computational performance. The study has been conducted using a Texas Instrument DSP commonly used in 3G base stations, and an open source linear programming solver. Although higher efficient commercial solvers are currently available, an open source solver has been used since access to its source code was needed for the JRRM implementation in the DSP platform. The computational cost of JRRM techniques using Texas Instrument DSP is first evaluated using the open source linear programming solver. The improvement that could be obtained with more efficient linear programming solvers is then estimated by means of computer simulations in order to have a more realistic JRRM computational cost estimation.

The rest of the paper is organized as follows. Next section presents the JRRM proposals that are analysed in this work, and evaluates their system and QoS performance. In Section 3, the hardware platform employed to investigate the computational cost of the proposed techniques is presented, and the linear programming tools employed to solve the JRRM dilemma are shortly described in Section 4. Section 5 presents the hardware computational cost of the JRRM policies, while Section 6 describes potential computational cost improvements.

2 JRRM Proposals

The implemented JRRM techniques, initially proposed in [10] and [11], are aimed at providing the highest possible homogeneous user satisfaction levels to all service types by exploiting the QoS/resource flexibility offered by different services present in a multimedia framework. For example, email users do not require the same number of radio resources than a video conferencing session to obtain the same user satisfaction levels. In this work, the user satisfaction is represented by utility values identifying the radio resources needed per service class to achieve certain user QoS satisfaction levels. To evaluate the efficiency of the implemented JRRM techniques, this work considers an heterogeneous wireless environment where the GPRS (General Packet Radio Service), EDGE (Enhanced Data rates for GSM Evolution) and HSDPA (High Speed Downlink Packet Access) RATs physically coexist.

The JRRM techniques have been implemented following the JRRM server approach discussed in the 3GPP standards ([1], [2]). This approach considers a centralized architecture that places the JRRM functionality in a node that collects information of all available RATs[1]. Since the JRRM techniques use utility functions to estimate the users' QoS demands, only updated information about each RAT's load must be transmitted to the JRRM server. Using this information, the implemented JRRM techniques manage the available radio resources to maximize the percentage of satisfied users. In particular, the JRRM policies have been designed to achieve optimum radio resource assignments following a user fairness approach: the proposed techniques aim at providing similar, and highest possible, utility levels for all service types, and only when the number of available radio resources is lower than the demand, will the implemented policy give priority to certain traffic classes. Both JRRM proposals are based on linear programming optimization techniques. To apply the linear programming mechanisms, the problem objective and constraints must be expressed as linear functions. Two different approaches have been proposed to achieve the sought problem objective. The optimal solution to both approaches is determined by the same system and service constraints.

2.1 Traffic Class Utility Values

Utility functions try to characterize the QoS satisfaction level experienced by a user based on the requested traffic service and the radio resources it has been assigned (combination of RAT and number of radio resources assigned within that RAT). This is a challenging task because user satisfaction is a subjective concept that heavily depends on user perceptions. The defined utility functions try to express the perceived user QoS as the link quality, and therefore data rate, varies. To establish the utility functions, the minimum, mean, and maximum QoS levels demanded by users are first defined per service class as illustrated in Fig. 1. This work considers a multimedia traffic scenario with email (background), web (interactive) and real-time H.263 video (with different mean bit rates) users.

For web and email services, utility values are expressed in terms of the user throughput. The minimum, mean and maximum QoS levels for web users have been defined as the throughput needed to satisfactorily transmit 90%, 95% and 97.5% of web pages in less than 4 seconds as established by the 3GPP TS 22.105 recommendations [13]. These high percentiles have been selected due the high transmission reliability requirements of non-real time data services. Web traffic is here modeled using the work reported in [14], whereas [15] has been used to model email traffic. The email model considers the transmission of emails with and without attachments, which makes it difficult to successfully transmit emails with large attachments within the 4 seconds 3GPP recommendations. Consequently, the email QoS thresholds have been established based on the throughput required to satisfactorily transmit 65%, 75% and 80% of the emails (with or without attachments). Once the QoS satisfaction thresholds have been established for web and email services, the utility functions have been defined so that users perceive a null utility value if their minimum QoS demand is not satisfied. This condition avoids assigning radio resources to users that would experience very poor QoS levels. Web and email user satisfaction linearly grows with the experienced throughput between the minimum and maximum QoS thresholds. Utility values equal to one have been avoided for web and email transmissions to account for the transmission reliability requirements of these services, and the dependence of the achievable throughput levels on the experienced channel quality conditions.

For real-time video services, video frames are considered to be satisfactorily transmitted if they are transmitted before the next video frame is to be transmitted. Consequently, the utility functions for real-time video services have been defined based on the percentage of correctly transmitted video frames, and the real-time video utility functions are independent of the mean video bit rates. The real-time video QoS satisfaction thresholds have been established considering the H.263 traffic model described in [16] and the indications provided in [17]. The studies reported in [17] show that a 25%, or even higher, dropping rate does not have a catastrophic effect on the QoS perceived by H.263 video users, and that dropping rates as high as 5% can be overcome if appropriate transmission techniques are invoked. Based on these results, the minimum and mean QoS satisfaction levels correspond to guaranteeing that 75% and 95% of video frames are transmitted before the next video frame needs to be transmitted. The maximum utility value for real-time video users has been set equal to one, and is achieved when all video frames are transmitted before the next video frame is to be transmitted. Although the 5% difference between the mean and maximum QoS levels might look negligible, this 5% includes the H.263 I-frames. These frames include information of independently coded

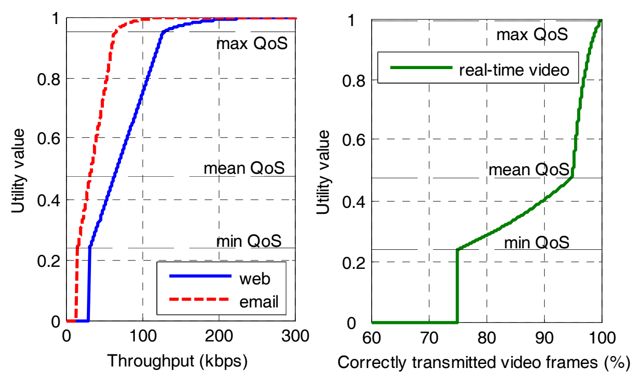

Fig. 1. Utility functions per traffic service.

[1] The implemented JRRM techniques could also operate under the 3GPP JRRM distributed architecture given the limited amount of information they require to be exchanged among different RATs (number of active users and their requested services).

images in a video sequence, and are also used to code/decode other images exploiting temporal redundancy. As a result, I-frames have a significant impact on the user perceived QoS level, and require high transmission rates due to their potential large size. Similarly to web and email users, real-time video users also perceive a null utility value below the minimum QoS threshold. Following the indications in [17] that highlight that an acceptable video quality requires a high percentage of correctly received video frames, the video utility increases slowly with the percentage of transmitted frames until the mean QoS level is achieved, and then rapidly until the maximum QoS level.

Once the utility functions are established, it is then necessary to relate the utility values with the different radio resource assignments. To establish this relation, the throughput achieved by each RAT and number of radio resources combination must be considered. However, it is difficult to estimate the throughput that could be achieved with a given number of radio resources given that the simulated radio access technologies implement link adaptation schemes. These schemes dynamically vary the used transmission mode (i.e., modulation and coding scheme) based on the experienced channel quality conditions. To account for these variations, and considering the difficulty to predict the achievable throughput in adaptive radio interfaces, the relation between the utility values and radio resource assignments has been established considering the data rate of the transmission modes providing a balance between high data rates and high error correction capabilities. In this context, average throughput values of 13.4 kbps and 22.4 kbps per timeslot (TS) have been selected for GPRS and EDGE, respectively, corresponding to the data rates of the coding scheme 2 (CS2) in GPRS, and the modulation and coding scheme 5 (MCS5) in EDGE [18]. In HSDPA, a high number of transmission modes are defined depending on the number of assigned codes. This work considers the transmission modes related to the 30 CQI (Channel Quality Indicator) values for User Equipment category 10 [19]. To achieve the sought balance between high data rate and high error correction capabilities, the selected transmission rate per number of assigned HSDPA codes corresponds to that achieved by the 'intermediate' transmission mode out of all possible modes for a given number of codes. Once the relation between throughput and radio resource assignment (combinations of RAT and number of radio resources) has been established, the utility values corresponding to each assignment can be obtained using the utility functions shown in Fig. 1.

For real-time H.263 video services, an additional step is necessary. A cumulative distribution function (CDF) of the throughput needed to transmit each video frame before the next video frame is to be transmitted is derived following the implemented H.263 video model [16]. Through these CDFs, the percentage of video frames reported in Fig. 1 can be related to the corresponding necessary throughputs for the various video bit rates considered in this work. Once the utility values are expressed as a function of the throughput, the utility values can be related to radio resources using the previously discussed relation between throughput and radio resources. Table 1 shows an example of the utility values obtained for the real-time 64 kbps H.263 video users with the different radio resource assignments; the utility values are listed according to the throughput provided by the corresponding RAT/radio resources combination. In this table, the assignments (RAT and number of radio resources) are denoted as xY, corresponding to x radio resources (timeslots or codes) from RAT Y (GPRS is represented as G, EDGE as E, and HSDPA as H). It is interesting to note that certain assignments cannot achieve utility values greater than zero.

TABLE 1
UTILITY VALUES FOR 64KBPSVIDEO USERS

Res./ RAT	Throughput (kbps)	Utility value	Res./ RAT	Throughput (kbps)	Utility value
1G	13.4	0.00	1H	116.5	0.38
1E	22.4	0.00	6E	134.4	0.44
2G	26.8	0.00	7E	156.8	0.93
3G	40.2	0.00	8E	179.2	0.98
2E	44.8	0.00	2H	396	1.00
4G	53.6	0.00	3H	741	1.00
5G	67	0.00	4H	1139.5	1.00
3E	67.2	0.00	5H	2332	1.00
6G	80.4	0.00	7H	4859.5	1.00
4E	89.6	0.29	8H	5709	1.00
7G	93.8	0.31	10H	7205.5	1.00
8G	107.2	0.35	12H	8618.5	1.00
5E	112	0.37	15H	11685	1.00

2.2 JRRM Policy Maximizing the Utility Values Homogeneously Assigned to Users

The first JRRM proposal seeks to maximize the multiplication of the utility values perceived by all the active users in the system, which results in the following objective function:

$$\max \prod_{j=1}^{N} u_j \tag{1}$$

which is equivalent to:

$$\max \ \ln \prod_{j=1}^{N} u_j \tag{2}$$

where u_j represents the utility value assigned to user j in a radio resources distribution round, and N corresponds to the total number of users in the cell. In scenarios where all users demand the same QoS and all radio resources offer equal QoS levels, (1) or (2) is satisfied when utility values are equally distributed among users [20]; the technique is thereby referred to as MAXIHU (MAXImum Homogeneous Utility values). On the other hand, it might not be possible to assign equal utility values to all users in multimedia scenarios with diverse and discrete radio resources. In this case, MAXIHU will try to maximize its objective function and homogeneously satisfy all users. MAXIHU's objective function can then be expressed as:

$$\ln \prod_{j=1}^{N} u_j = \sum_{j=1}^{N} \ln u_j \tag{3}$$

with u_j defined in (4):

$$u_j = \sum_{r=1}^{3} \sum_{s=1}^{c^r} U_j(r,s) \cdot y_j^{r,s} \tag{4}$$

In (4), $U_j(r,s)$ represents the utility value obtained by user j when assigned s radio resources (codes or timeslots) of RAT r (r is equal to 0, 1 or 2 for GPRS, EDGE and HSDPA respectively), and $s \in [1,c^r]$ with c^r corresponding to the maximum number of radio resources available at each RAT. $y_j^{r,s}$ is a binary variable equal to one if user j is assigned s radio resources of RAT r, and equal to 0 if not. The proposed JRRM policy focuses then on deciding for each user which $y_j^{r,s}$ variable is equal to one, considering that only $y_j^{r,s}$ variables achieving a utility value greater than zero are allowed. Given that only one $y_j^{r,s}$ variable can be equal to one for each user, the following expression applies:

$$\sum_{j=1}^{N} \ln u_j = \sum_{j=1}^{N} \ln \sum_{r=1}^{3} \sum_{s=1}^{c^r} U_j(r,s) \cdot y_j^{r,s} = \sum_{j=1}^{N} \sum_{r=1}^{3} \sum_{s=1}^{c^r} \ln\left(U_j(r,s) \cdot y_j^{r,s}\right) \tag{5}$$

To express the objective function as a lineal equation, all users must have a variable $y_j^{r,s}$ equal to one. As a result, (5) becomes:

$$\sum_{j=1}^{N} \sum_{r=1}^{3} \sum_{s=1}^{c^r} \ln\left(U_j(r,s) \cdot y_j^{r,s}\right) = \sum_{j=1}^{N} \sum_{r=1}^{3} \sum_{s=1}^{c^r} \ln\left(U_j(r,s)\right) \cdot y_j^{r,s} \tag{6}$$

MAXIHU objective function can then be expressed as:

$$\max \sum_{j=1}^{N} \sum_{r=1}^{3} \sum_{s=1}^{c^r} \ln\left(U_j(r,s)\right) \cdot y_j^{r,s} \tag{7}$$

2.3 JRRM Policy Maximizing the Minimum Utility Value Perceived by a User

The second approach seeks to maximize the lowest utility value assigned to a user in a radio resources distribution round. This policy increases the minimum QoS that can be perceived by any user in the system, and is thereby referred to as MAXILOU (MAXImise Lowest Utility). MAXILOU's objective function can then expressed as:

$$\max \quad \min_{j \in \{1,\ldots,N\}} u_j \tag{8}$$

where u_j is defined in (4). In order to apply linear programming techniques to solve the established problem, (8) must be expressed as a linear equation. To this aim, a new real variable denoted z, and equal to the smallest utility value assigned to a user, has been defined, which results in the following objective function:

$$\max \ z, \quad \text{with} \quad z \le u_j \quad \forall j \in \{1,\ldots,N\} \tag{9}$$

2.4 JRRM Constraints

Once the JRRM objective functions have been defined, the problem statement must be completed with the system and service constraints. The first system constraint is conditioned by the limited number of available radio resources in the system (10). When such limitation prevents the possibility to grant all users their minimum QoS demand, none of the two JRRM proposals would result in a satisfactory solution. In fact, MAXIHU's objective function does not even consider this possibility (see Section 2.2). In the case of MAXILOU, its objective function is equal to the null value when there are not enough resources to satisfy the minimum QoS demand to all active users. In that case, whatever radio resource distribution with at least one user perceiving the zero utility value is an optimum solution to the linear programming problem. To avoid this situation, the system constraint (11) imposes that one $y_j^{r,s}$ variable must be equal to one for each active user. Since (11) would be unfeasible if there are not enough resources to satisfy the minimum QoS level to all active users, some users should be eliminated from the radio resources distribution process.

$$\sum_{j=1}^{N} \sum_{s=1}^{c^r} s^r \cdot y_j^{r,s} \le c_r, \forall r \in \{1,2,3\} \tag{10}$$

$$\sum_{r=1}^{3} \sum_{s=1}^{c^r} y_j^{r,s} = 1, \forall j \in \{1,\ldots,N\} \tag{11}$$

MAXILOU and MAXIHU base their resource distribution decisions on the load conditions and the different users/services QoS requirements. As a result, this work applies the JRRM mechanisms whenever a transmission ends or a user requests resources for a new transmission. In this case, only active video users that were assigned resources in the previous JRRM distribution round can maintain the minimum number of their assigned resources ($s_{min,j}$ radio resources from RAT $r_{min,j}$) that guarantees their minimum QoS demand. These video users have then to compete with the rest of users for other radio resource assignments further improving their QoS satisfaction. This condition can be expressed as follows:

$$\sum_{r=1}^{3} \sum_{s=1}^{c^r} U_j(r,s) \cdot y_j^{r,s} \ge U_j(r_{\min,j}, s_{\min,j}) \quad \forall j \in \{1,\ldots,N \mid t_j = 3\} \tag{12}$$

where t_j represents the traffic type demanded by user j (t_j is equal to 1, 2 or 3 for email, web and real-time video services respectively).

In scenarios where it is not possible to achieve equal utility values for all active users due to the scarcity of available radio resources, users are served based on the following service priority: real-time H.263 video (higher priority), web, and email. Among real-time video users, those with higher mean video bit rates are served first. The user priority criterion is represented by $\succ$ ($k \succ j$ indicates that user k is higher priority than user j). When the service prioritization criterion is applied between two video users characterized by different mean video bit rates, the lowest priority user (m) could also have obtained radio resources in the previous JRRM distribution round. In this case, the condition established in (12) comes first, and user m will maintain the $s_{min,j}$ radio resources from RAT $r_{min,j}$ needed to guarantee its minimum QoS level. When such minimum level is achieved, the lowest priority user will not be assigned additional resources until the highest priority user (k) surpasses its utility value ($U_j(r_{\min,j}, s_{\min,j})$). This constraint is expressed as:

$$\sum_{r_a}\sum_{s_a} U_j(r_{\min,j}, s_{\min,j}) \cdot y_k^{r,s} + \sum_{r_b}\sum_{s_b} U_k(r,s) \cdot y_k^{r,s} \geq \sum_{r=1}^{3}\sum_{s=1}^{c^r} U_j(r,s) \cdot y_j^{r,s} \quad (13)$$

$$\text{if } k \succ j, \forall k, j \in \{1,\ldots,N\} \text{ and } k \neq j$$

where (r_a,s_a) represents the subset of the total possible RAT/resource assignments (r,s) that verify $U_k(r,s) < U_j(r_{\min,j}, s_{\min,j})$, and (r_b,s_b) the subset of the total possible RAT/resource assignments that verify $U_k(r,s) \geq U_j(r_{\min,j}, s_{\min,j})$. Following (13), if active users cannot obtain their minimum QoS demand (it is not possible to satisfy (10) and (11)) and the linear objective function does not have a solution, users with the lowest priority will be eliminated from the JRRM distribution round until the present users and their respective demands allow for a linear programming JRRM solution.

2.5 JRRM System Performance

The performance of MAXILOU and MAXIHU has been evaluated in a simulation platform that emulates the distribution of GPRS, EDGE and HSDPA radio resources among real-time H.263 video, email and web users. It has been simulated one frequency carrier per RAT, i.e. eight timeslots for GPRS and EDGE, and 14 HSDPA codes. In terms of service distribution, email, web and real-time video transmissions represent each a third of the new service requests; new video service requests are equally distributed among 64, 256 and 512kbps video bit rates. All three RATs are assumed to provide the same radio coverage. Fig. 2 depicts the utility values per service class achieved by each JRRM proposal considering 10 and 20 users per cell respectively. The figure shows the percentage of users per service class that achieve the utility values corresponding to the minimum, mean and maximum QoS levels shown in Fig. 1. The simulated scenarios result in a traffic load higher than the load that could be served with the simulated radio resources. In this context, it is not possible for the simulated scenarios that all service classes achieve maximum QoS levels. However, the obtained results show that both JRRM proposals satisfy their various objectives:

- The majority of services achieve their minimum QoS level, and only when such level is guaranteed, resources are additionally assigned to higher priority users.
- The number of served users is the maximum possible satisfying the system and service constraints.
- The service priorities criterion defined in (13) is correctly applied under radio resources shortage conditions.

It is also important to highlight that MAXILOU and MAXIHU assign the same utility value to the user that perceived the lowest utility value at each radio resources distribution round, which means that both proposals achieve the maximum possible utility value for that user. However, a higher percentage of users perceive higher QoS levels when MAXIHU is applied as a result of the different objectives functions. The implementation of MAXILOU guarantees that the highest possible minimum utility value assigned to any user is reached. However, when this objective is fulfilled, the other users with higher utility values stop competing for additional radio resources that could further improve their QoS satisfaction

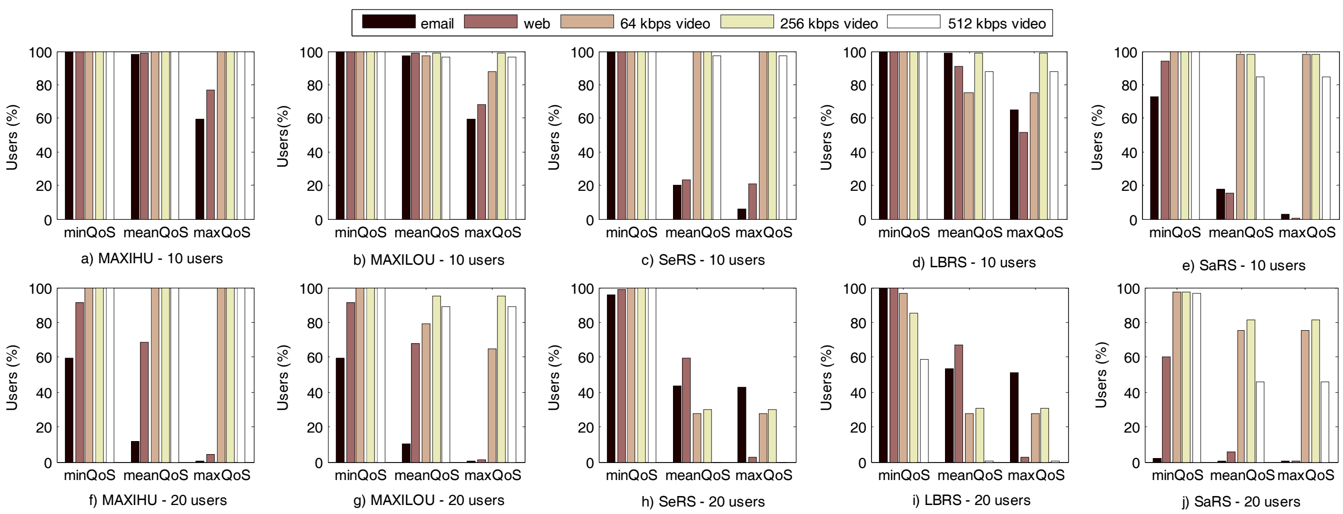

Fig. 2. Achieved utility values per service class.

level. This is the case because when the minimum utility value assigned to any user is maximized, MAXILOU terminates its JRRM radio resources distribution round. On the other hand, if MAXIHU is applied, the remaining users still compete for additional resources to improve their perceived utility value since the objective function increases as the users' utility values increase. This situation is also highlighted when analyzing the percentage of JRRM distributions over which radio resources are left unassigned. While MAXILOU left unassigned radio resources in 81% and 42% of the JRRM distribution rounds in the 10 and 20 users per cell scenarios respectively, these percentages decrease to only 39% and 11% for MAXIHU. These results clearly highlight MAXIHU's more efficient use of the available radio resources.

An important characteristic of heterogeneous wireless systems is the possibility to conduct vertical handovers (VHO) between RATs. Although such handovers can increase the QoS, they also incur in an additional delay and overhead that must be carefully controlled, in particular for delay sensitive real-time services. The time required to execute vertical handovers has been measured in real mobile networks, and the obtained VHO delays for voice calls are shown in Table 2 (the VHO is conducted from UMTS to GSM). To perform the measurements, the Nokia 6720c handset supporting GSM/GPRS/EDGE and UMTS/HSDPA has been used. The engineering mode terminal incorporates the Nemo Handy application, which provides the terminal with a powerful radio monitoring capability. Nemo Handy provides extensive network parameters and exchanged signaling messages captured over voice calls and data transfers. The logged measured data has been processed using the Nemo Outdoor software tool. As shown in Table 2, the vertical handover procedure currently implemented in mobile networks only resulted in an average delay of 157 ms, which is tolerable for voice services. The vertical handover procedure for data services was also evaluated. In this case, it is important to note that although a packet switched (PS) vertical handover procedure is already defined in the 3GPP specifications [21], the measured mobile network didn't implement it since it was based on 3GPP Release 5. Instead, it used a cell reselection procedure to switch RATs for active PS users, which increased the inter-RAT change delay to a few seconds. Such increase will be avoided when the defined 3GPP PS vertical handover procedure (3GPP Release 6) is implemented in mobile networks; in fact, the 3GPP standards indicate in [22] that a PS vertical handover from GSM to UMTS/HSDPA cells must be executed in less than 220 and 190 ms for FDD and TDD cells respectively (assuming good radio conditions [22]).

Despite the short measured VHO delays, vertical handovers must be controlled, in particular for real-time services with tight delay requirements. To this aim, the MAXIHU and MAXILOU proposals guarantee that in each distribution round, active real-time video users will maintain at least their minimum QoS level using resources from the RAT they were previously assigned (12). These users will only change RATs if they can obtain higher QoS levels using resources available from other RATs. This approach has been adopted to achieve a balance between QoS and cost of VHOs. For non real-time services, vertical handovers are permitted without any restrictions due to their higher tolerance to delays. Fig. 3 shows the percentage of transmissions that ended up with and without switching RATs for real-time services. This figure confirms that the two JRRM proposals limit the number of vertical handovers for real-time services. For non-real time services, the percentage of sessions that didn't experience a VHO is reduced to 53% and 24% with MAXIHU, and to 51% and 24% with MAXILOU when considering 10 and 20 active users per cell respectively. The increase in the number of VHOs for non-real time users as the load increases is justified by the QoS benefits that such VHOs produce (Fig. 2).

Finally, the performance of MAXIHU and MAXILOU is compared to some well established JRRM techniques reported in the literature:

- Service based RAT selection, SeRS [3]. This technique is based on pre-established service-to-RAT assignments. For each service, a prioritized list of RATs is maintained. When a new user requests access to the system, the system tries to allocate the user to the first RAT from its list with available capacity.
- Load balancing based RAT selection, LBRS [23]. The LBRS technique assigns each user requesting access to the system to the RAT having the lowest load. The load metric is calculated as the ratio of utilized capacity to the total available capacity in each RAT.
- Satisfaction based RAT selection, SaRS [24]. Each time a new user requests access to the system, this technique evaluates the number of satisfied users in each RAT, and assigns the new user to the RAT with a higher percentage of satisfied users.

The QoS levels obtained by each of the reference tech-

TABLE 2
TIME TO EXECUTE A VHO FOR VOICE CALLS

Average	157 ms
Minimum	58 ms
Maximum	274 ms

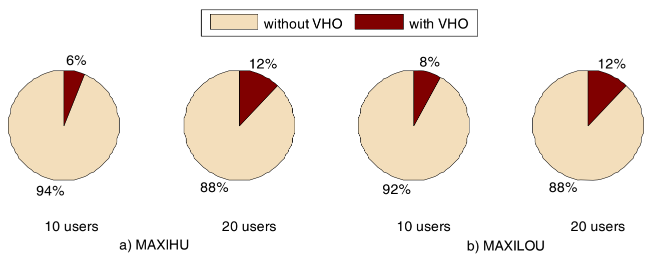

Fig. 3. Percentage of real-time video transmissions that ended up with/without switching RATs.

niques are also depicted in Fig. 2. The results clearly show that MAXIHU and MAXILOU outperform the three reference techniques in the simulated scenarios. Only in the scenario with 20 users per cell, LBRS and SeRS achieve higher QoS levels for the lowest priority users, but this is done at the expense of significantly degrading the QoS performance for the rest of services. This is due to an inefficient LBRS and SeRS resources distribution that resulted in low priority or background users being assigned resources with transmission capabilities exceeding their QoS demands. As a result, these services achieve higher QoS satisfaction levels than real-time video users. These results highlight MAXIHU and MAXILOU's high QoS performance, as well as their capacity to adapt and satisfy the established system conditions and QoS objectives under varying operating conditions. Although the reference techniques have a lower computational cost, the next section will demonstrate the implementation feasibility of the JRRM proposals based on linear programming and optimization techniques. In this context, and taking into account the QoS limitations of the reference techniques, the JRRM proposals are characterised by a favourable performance versus computational cost trade-off.

3 Hardware Platform

Cellular base stations must continuously handle increasing capacities, process higher data rates, and support multimedia standards, while at the same time there is an increasing demand for reduced size, cost and power consumption of communications equipment. In this context, the evolution of cellular technologies is highly dependent on the evolution and adoption of high performance DSPs. The TMS320C6000™ DSP architecture of Texas Instruments is capable of scaling to speeds faster than 1 GHz, and achieves around 9000 MIPS for single-core devices [25]. The TMS320C6455 DSP is one of the highest-performance fixed-point DSP in the TMS320C6000™ DSP platform, and it is the one used in this work to estimate the computational performance of the proposed JRRM algorithms. It performs at up to 9600 MIPS at a 1200 MHz clock rate, and works with a 32 bit word enabling a high accuracy in arithmetic operations [26]. The TMS320C6455 DSP core employs eight functional units to achieve maximum parallelism in processing 3G algorithms, each of them capable of executing one instruction every clock cycle [26].

The computational performance of the proposed JRRM techniques has been evaluated using the Code Composer Studio (CCStudio) software [27]. This software is the integrated development environment for Texas Instrument's (TI) DSPs, and includes compilers for each of TI's device families. This software also incorporates a tool that enables the real-time simulation of most of TI's DSPs, including the TMS320C6455. The CCStudio also includes source code editor, project build environment, debugger and profiler features. These tools enable users to produce an efficient code for their applications employing C or C++ programming language. In the debug session, the C/C++ code or the corresponding machine code can be shown and also the value of memory positions, registers, or variables can be monitored. Furthermore, CCStudio's interactive profiler provides some program performance analysis parameters, such as the number of elapsed clock cycles, which enable measuring the computational cost of the application that is being executed.

4 Linear Programming Resolution

4.1 Linear Programming Mechanisms

The problem statement, system and service constraints have been mathematically defined in Section 2. The problem objective has been expressed by a linear objective function with binary integer unknown variables $y_j^{r,s}$, and also a real variable z in the case of MAXILOU. In operations research, the type of problems that only consider binary integer variables is referred to as Binary Integer Programming (BIP), while those also considering some real variables are referred to as Mixed Integer Programming (MIP) [28]; BIP problems are a particular case of MIP problems and the same methods are employed to solve them. Different approaches can be used to solve MIP problems. One of the most popular approaches due to its performance and computational properties is the Branch and Bound (B&B) method [28]. This technique solves an ordered sequence of reduced linear programming problems until an optimum solution is achieved. The sequence of problems is obtained by reducing the possible set of values for every integer variable. Each reduced problem is referred to as a node since the B&B resolution method is usually represented by a tree topology. When a binary variable is considered, two reduced problems can be derived by fixing the variable value to zero or one. To solve these reduced problems, the integer condition of the unknown variables is relaxed, and real values are allowed. The simplex method is then applied to the resulting linear programming problems [28]. The simplex method is regularly employed in linear programming problems with a large number of variables that require computationally efficient solutions. The simplex method is an algebraic procedure that makes use of the fact that the linear functions expressing the system and user constraints present in the problem statement reduce the range of possible solutions to a limited spatial region. It has been demonstrated that the solution that optimizes the objective function is placed in a vertex of this region [28]. Therefore, the simplex method moves from one vertex to another one improving the objective function value until no better solution can be obtained. In this context, an iteration of the simplex method is made for each evaluated vertex.

Since the Branch and Bound method was proposed, more efficient and faster methods have been developed to solve MIP problems. For example, the Branch and Cut method [28] incorporates the use of cutting planes to the

B&B method. Cutting planes are new functional constraints that reduce the feasible solutions region of the relaxed linear programming problem without eliminating feasible solutions to the original MIP problem. The Branch and Cut method has therefore been employed in this work to solve the MIP problems.

4.2 Linear Programming Solvers

To solve the linear programming problems associated to the radio resources distribution dilemma investigated in this work, two different linear programming software applications have been considered: a state-of-the-art commercial solver called CPLEX [29], and LP_SOLVE, a commonly-used open source solver [30]. Both linear programming solvers implement the required mechanisms to solve the MIP problems discussed in the previous section. CPLEX is a powerful software that incorporates the fastest and most efficient fundamental algorithms to solve mathematic optimization problems with high computational requirements. Regarding MIP resolution mechanisms, CPLEX employs state-of-the-art algorithms and techniques as well as proprietary solutions to solve difficult MIP problems. Despite its high performance, CPLEX is a commercial solver and it is then not possible to access its source code, which is required to evaluate its computational performance in the DSP emulator software. In this context, LP_SOLVE, which is an open source solver, has also been considered in this work given that its JRRM computational performance can be studied with the CCStudio software. LP_SOLVE is released under the LGPL (the GNU lesser general public license) license, and many people have contributed to its development.

Several studies have previously evaluated the performance of both solvers. For example, Brglez and Osborne [31] showed that a solver's computational performance depends on the problem statement format, for example on the order at which variables and constraints are expressed in the instance of the problem. According to [31], very different computational performance results can be achieved with several instances of the same problem on the same platform and with the same version of the solver. Consequently, a high number of problems should be solved and statistical data should be provided. The time spent by CPLEX and LP_SOLVE to obtain the optimal solution for linear programming problems has been compared in [32]. This work showed that CPLEX is 100 times faster than LP_SOLVE executing the simplex method to solve the selected problems. The performance of both solvers addressing MIP problems has also been measured in [33] and [34]. These studies provide the CPU and user solution times respectively for a wide variety of problems, demonstrating the higher execution time efficiency of the CPLEX solver. The results depicted in [34] show that only in less than 0.03% of the considered problems, the elapsed time is similar for both solvers. For the remaining problems, CPLEX outperforms the results obtained by LP_SOLVE, and while CPLEX solves a high number of problems in less than 1 minute, LP_SOLVE does not even achieve the optimum solution for those problems when a time limit of 2 hours is considered. Despite the higher CPLEX efficiency, it can not be implemented in the DSP emulator application due to its commercial nature. As a result, LP_SOLVE is the linear programming solver used in the CCStudio DSP software simulator to solve the JRRM problems.

5 Computational Performance Analysis

To evaluate the applicability of the proposed JRRM algorithms in real systems, this work has implemented them in the TMS320C6455 DSP using the DSP simulator CCStudio (Section 3). The profiling tool of the DSP simulator provides the number of elapsed clock cycles for each function in the program. The time required to solve a JRRM problem can then be calculated dividing the number of elapsed clock cycles by the frequency of the internal clock (1200 MHz for the TMS320C6455 [26]). As previously discussed, the linear programming software used in the DSP simulator to solve the MIP JRRM problems is the LP_SOLVE 5.5 solver (Section 4).

Fig. 4 depicts the computational time required by MAXIHU and MAXILOU to find a JRRM distribution solution when implemented in the TMS320C6455 using LP_SOLVE. The results correspond to a scenario where email, web and real-time video transmissions represent each a third of the new service requests, and new real-time video service requests are equally distributed among 16, 64 and 128kbps video bit rates. Given that no time threshold has been currently defined in the community to determine whether a JRRM algorithm is feasible or not feasible, this work considers as valuable benchmarks the time needed in current mobile networks to assign radio resources to a new user or conduct a vertical handover. In this context, it is important to highlight that active users do not stop or pause their transmissions while the JRRM algorithm is being executed since the algorithm is based on utility functions previously derived in an offline process. Since the JRRM techniques are executed each time a user requests access to the system or ends its transmission, the JRRM execution time would only have an effect on new users. Field measurements have been conducted to measure the time needed to assign radio resources to a new user or conduct a vertical handover. The conducted measurements have shown that 3 to 6 seconds are needed to assign radio resources to a new user (validating the indications reported in [35]), while vertical handovers require an average of 157ms for voice transmissions (Table 2)[2]. Taking into account these measurements, JRRM execution times of some hundreds milliseconds can be considered reasonable times to validate the possible implementation of the proposed JRRM policies in mobile networks.

Based on the previous reasoning, the results depicted

[2] The average time needed to conduct a vertical handover increased to 6.7 seconds for data transmissions in current networks.

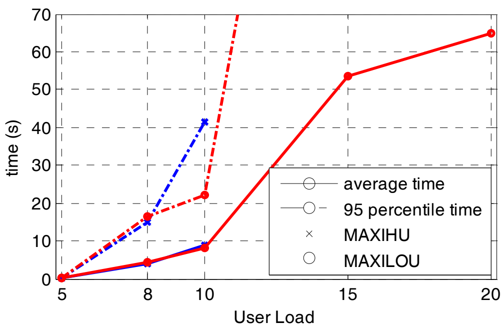

Fig. 4. MAXIHU and MAXILOU real-time computational performance (in seconds): TMS320C6455 and LP_SOLVE.

in Fig. 4 show that MAXILOU is capable to find an optimum JRRM solution when 5 users actively request resources in reasonable times: the average time needed to find a JRRM distribution solution is equal to 0.21s, whereas 95% of the JRRM problems were solved in less than 0.26s. Similarly to MAXILOU, the time needed by MAXIHU to find an optimum JRRM solution using LP_SOLVE seems only viable for scenarios with 5 users (the average time is equal to 0.09s, whereas 95% of the JRRM problems were solved in less than 0.11s). The performance for both techniques degrades as the user load increases, although MAXILOU required in average lower execution times than MAXIHU for higher loads. The inefficiency of some LP_SOLVE algorithms prevented analysing the MAXIHU execution times in the DSP platform for scenarios with 15 and 20 users per cell (DSP runs out of memory). To analyse these scenarios, simulations have been conducted using a PC with a 2.6GHz AMD Opteron processor, 1MB of cache and 3GB of RAM. The results depicted in Table 3 show that the CPU time needed by MAXIHU to find optimum JRRM distribution solutions is higher than required by MAXILOU, and significantly increases for loads above 10 users per cell. The degraded MAXIHU computational performance is due to the fact that the simplex and Branch and Cut implementations in LP_SOLVE are not adequate to solve the MAXIHU MIP problems when a high number of users simultaneously demand radio resources.

The results depicted in Fig. 4 and Table 3 seem to indicate that the proposed JRRM algorithms do not achieve acceptable execution times when a high number of users are participating in the distribution process. However, it is important to remember that these results have been obtained using LP_SOLVE, and Section 6 will demonstrate that significant improvements can be achieved with a more efficient MIP solver.

6 Improving Computational Performance

6.1 Optimized source code

The MAXIHU and MAXILOU execution times shown in Fig. 4 have been obtained from the number of elapsed clock cycles provided by the CCStudio DSP simulator. This DSP simulator also provides the number of executed instructions. Although the TMS320C6455 is able to perform at 9600 MIPS using in parallel its 8 available functional units, the analyses of the number of executed instructions shows that, in average, only one instruction is executed every 5 or 6 clock cycles depending on the number of users participating in the resources distribution process. This low number of executed instructions per clock cycle is due to the fact that the source code of the JRRM algorithm and the linear programming tools has not been optimized to be implemented in the DSP platform. The non-optimized source code results in a high number of clock cycles spent without executing instructions due to cache penalties and/or memory wait states required by the physical device to access memory and read data. It is important to highlight that the linear programming solver employed in this work is developed to be implemented on a computer, and is usually applied to analyse and solve problems where the time required to access memory is not a critical issue. In this context, a computational improvement factor of up to 40 or 48 could be achieved with an optimized code that utilizes the eight functional units available in the TMS320C6455.

TABLE 3
CPU Time (in Seconds)

		MAXIHU				MAXILOU			
Users per cell		8	10	15	20	8	10	15	20
LP_SOLVE	Avg	0.06	0.88	201	868	0.06	0.28	4.04	9.72
	95perc	0.27	3.96	1090	6897	0.16	0.93	9.02	15.39
CPLEX	Avg	0.01	0.03	0.21	0.16	0.02	0.04	0.23	0.48
	95perc	0.02	0.09	0.46	0.47	0.03	0.09	0.83	0.55

6.2 Linear Programming Solver

As discussed in Section 4.2, the CPLEX solver has been shown to be more computationally efficient than LP_SOLVE in finding optimum solutions to MIP problems. Consequently, it is worthwhile analyzing the computational improvement that could be obtained if CPLEX was used instead of LP_SOLVE to implement the MAXIHU and MAXILOU JRRM proposals. Due to the unavailability of CPLEX source code for the DSP implementation, the CPLEX computational improvements have been evaluated by means of computer simulations executed on a 2.6GHz AMD Opteron processor. Although this evaluation environment does not provide a direct indication of the computational performance on a DSP hardware platform, it provides useful information about the improvements that could be obtained if more powerful solvers were used in the DSP implementation. The computational comparison has been conducted using the CPLEX 9.1.0 and LP_SOLVE 5.5 versions.

Table 3 compares the CPU time required by LP_SOLVE and CPLEX to solve the MAXIHU and MAX-

ILOU JRRM resource distribution problems under different cell loads. The conducted simulations revealed that only when 5 users actively request resources, LP_SOLVE is faster than CPLEX, although both solvers achieve the optimum solution in very short times. LP_SOLVE requires average times equal to 0.7ms and 1ms to achieve the optimum solution with MAXIHU and MAXILOU, while CPLEX required on average 2ms for both JRRM techniques. On the other hand, CPLEX significantly reduces the time needed to find JRRM solutions, in particular under high cell loads where MIP problems with a large number of variables and constraints need to be solved. This trend is also observed in Fig. 5, which represents the CDF of the time required by each solver to achieve a JRRM solution when MAXIHU is applied in scenarios with 10 and 15 users per cell. The benefits obtained using CPLEX vary based on the simulated scenario and the evaluated JRRM technique. For example, CPLEX solves 95% of the JRRM problems with 8 users demanding resources in less than 22ms and 31ms when MAXIHU and MAXILOU are applied, while this percentage is reduced to 53% and 34% when LP_SOLVE is used. With 10 users per cell, the 95 percentile time is reduced from 3.96s and 0.93s for MAXIHU and MAXILOU using LP_SOLVE to just 89ms and 90ms with CPLEX. In this context, it is worthwhile noting that CPLEX achieves similar MAXIHU and MAXILOU computational times, which was not the case when using LP_SOLVE. Even if the cell load is further increased, CPLEX is still capable to guarantee execution times of just a few hundred milliseconds, with reductions of approximately one order of magnitude when comparing MAXILOU's 95 percentile. Based on the obtained results, the use of CPLEX, together with an optimized source code, will provide acceptable execution times for the implementation of MAXIHU and MAXILOU in real systems. As an example, Table 4 shows estimated MAXILOU DSP execution times when the CPLEX and optimized source code improvements previously reported are applied to the MAXILOU DSP execution times measured using LP_SOLVE and reported in Fig. 4. The comparison of Table 4 and Fig. 4 shows that while MAXILOU needed on average 48 seconds to find an optimum resources distribution when 15 active users requested radio resources and LP_SOLVE was used, this value would be reduced to just 3 seconds if CPLEX is used instead. Section 6.1 indicated that a computational improvement factor of up to 40 or 48 could be achieved with an optimized code that utilizes the eight functional units available in the TMS320C6455. If we just consider an optimized code improvement factor of 10 together with the use of CPLEX, Table 4 shows that MAXILOU's implementation in DSP platforms would just require around 300ms to distribute radio resources among 15 active users per cell.

TABLE 4
ESTIMATED AVERAGE MAXILOU DSP EXECUTION TIME (IN SECONDS)

Users per cell	8	10	15	20
Applying CPLEX improvement factor	1.34	1.23	3.07	3.21
Applying CPLEX and optimized source code improvement factor	0.14	0.13	0.31	0.32

The computational differences observed with LP_SOLVE and CPLEX are due to their different methodologies to implement and execute the simplex and Branch and Cut methods (section 4.1). Fig. 6 represents the CDF of the total number of iterations executed by the simplex mechanism, and Fig. 7 the CDF of the number of nodes explored by the Branch and Cut method until the optimal solution to the JRRM problem is found when MAXIHU is applied. Both figures show that the simplex and Branch and Cut implementations are more efficient in the case of CPLEX than in the case of LP_SOLVE. For

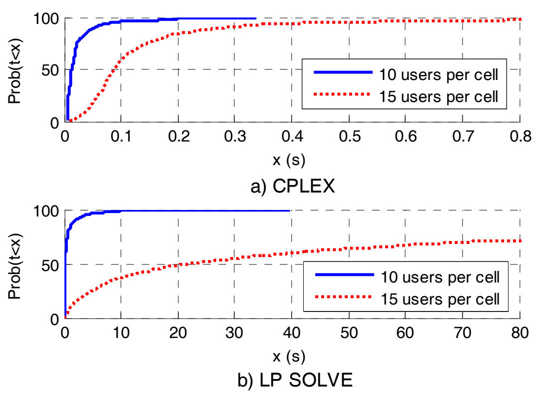

Fig. 5. CPU time required MAXIHU with the CPLEX and LP_SOLVE solvers.

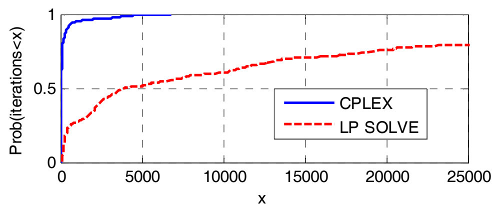

Fig. 6. CDF of the number of iterations executed by the simplex method for 10 users per cell.

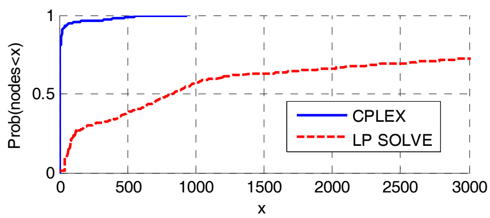

Fig. 7. CDF of the number of nodes explored by the Branch and Cut method for 10 users per cell.

example, LP_SOLVE requires 54.6 times more iterations and explores 284.4 times more nodes than CPLEX for 50% of analyzed JRRM problems.

Finally, it is also relevant noting that better performance could even be achieved if a solver designed to consider the specific characteristics of the MIP JRRM problems could be used instead of a general purpose linear programming solver such as CPLEX [31]. This fact justifies the better performance of LP_SOLVE when a low number of users participate in a JRRM distribution process; the methodologies implemented in LP_SOLVE focus on MIP problems with a relatively low number of variables and constraints. On the other hand, the methodologies implemented in CPLEX are good for a wide type of MIP problems, which results in good computational performance independently of the number of variables and constraints.

6.3 Suboptimal Solutions

Previous sections have demonstrated the feasibility of implementing the proposed JRRM techniques in real mobile communication systems using powerful hardware and software tools. The computational execution cost can be further reduced at the cost of eliminating the optimality condition in the radio resources distribution. In this context, the computational performance and the QoS satisfaction levels should be evaluated to achieve a suitable tradeoff between both parameters.

To reduce the time needed to solve a JRRM problem, a variant of the MAXIHU technique is here analysed. The variant ends the radio resources distribution process when a feasible solution previous to the optimal one is achieved, and this suboptimal solution satisfies a given condition. This condition refers to the gap between the current feasible solution a, and the solution b corresponding to the optimum objective value achieved for the JRRM problem when the integer condition of all the unknown variables is relaxed and real values are allowed (b is the bound solution to the JRRM problem). The gap between both solutions is calculated as follows:

$$gap_{a-b}(\%) = \text{abs}\left(\frac{a-b}{1+\text{abs}(b)}\right) * 100 \tag{14}$$

where abs() represents the absolute value. The suboptimal condition was established so that when the gap between both solutions a and b is lower than 25%, the JRRM algorithm stops and adopts the suboptimal solution (a) as its JRRM solution to the radio resources distribution process under study. The suboptimal approach was tested through computer simulations for 10 users per cell. The tests resulted in that 38.46% of the JRRM problems explore, in average, 313 nodes less than the optimum solution achieved with the original JRRM MAXIHU technique. However, this implementation cost reduction didn't reduce significantly the computation time or the user QoS level.

The simulations conducted in Sections 5 and 6.2 showed that MAXIHU's JRRM resolution process was capable to rapidly find and improve feasible, but not optimal, solutions. After this initial phase, the JRRM process only improves slightly and slowly the objective function despite exploring a high number of nodes. This trend emphasizes a possible trade-off between performance and implementation cost, since it is possible to find a suboptimal solution with a much smaller computational cost. In this context, a second condition is applied during the JRRM resolution process to try to reduce the computational time: if the gap between the current suboptimal feasible solution and the previously achieved feasible solution is lower than 10%, the algorithm ends its radio resources distribution process. When this approach was applied in the scenario with 10 active users per cell, 86.42% of the JRRM problems explored, in average, 4582 nodes less than if the optimum solution was achieved. Fig. 8 represents the CDF of the time required by the JRRM MAXIHU algorithm using LP_SOLVE to achieve the optimum solution, and the time needed to achieve the suboptimal solution when both suboptimal conditions are applied. This figure corresponds to scenarios where 8 and 10 users per cell demand radio resources. The percentage of users per service class that achieve the minimum, mean and maximum QoS levels shown in Fig. 1 when the optimum and suboptimum JRRM processes are applied is reported in Tables 5 and 6, respectively. The results obtained show that the computational cost of advanced JRRM techniques can be significantly reduced with suboptimal solutions that do not significantly degrade the user perceived QoS levels. The conducted study has shown that, under the evaluated conditions, the QoS degradations resulting from the non-optimality JRRM approach only affect the less priorised services, while the most demanding services are capable to maintain their maximum QoS satisfaction levels.

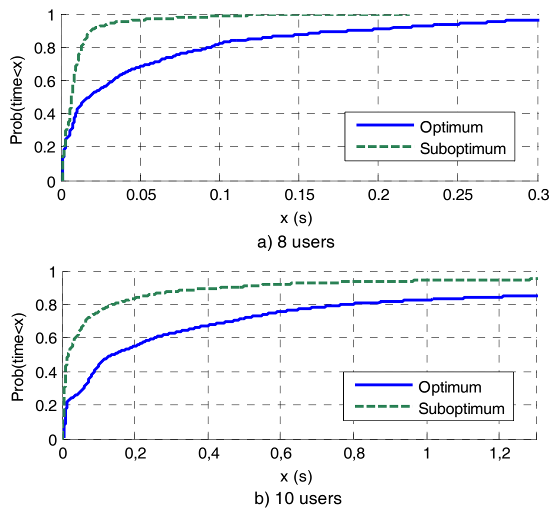

Fig. 8. CPU time required by the optimum and suboptimum JRRM resolution processes.

TABLE 5
QOS LEVELS (%) PER SERVICE CLASS FOR 8 ACTIVE USERS PER CELL

	Optimum			Suboptimum		
	min QoS	mean QoS	max QoS	min QoS	mean QoS	max QoS
Email	100	100	99.94	100	99.99	99.94
Web	100	100	99.81	100	100	99.68
16kbps video	100	100	100	100	100	99.97
64kbps video	100	100	100	100	100	100
128kbps video	100	100	100	100	100	100

TABLE 6
QOS LEVELS (%) PER SERVICE CLASS FOR 10 ACTIVE USERS PER CELL

	Optimum			Suboptimum		
	min QoS	mean QoS	max QoS	min QoS	mean QoS	max QoS
Email	99.98	99.97	87.98	100	92.59	76.52
Web	100	100	96.11	100	100	93.85
16kbps video	100	100	100	100	100	99.94
64kbps video	100	100	99.93	100	100	98.93
128kbps video	100	100	100	100	100	100

7 CONCLUSIONS

This work has conducted the first hardware implementation feasibility study of advanced JRRM techniques for heterogeneous wireless networks. The proposed, implemented and evaluated techniques are based on linear programming and optimization algorithms, and have been shown to achieve good system performance under multimedia traffic conditions. To evaluate their implementation feasibility, the JRRM techniques have been implemented in a DSP simulator software using open source linear programming solvers. The conducted study has shown the feasibility to implement these novel policies in real systems using state-of-the-art chipsets that allow achieving acceptable JRRM execution time under medium loads and a large number of variables. The study has also revealed that the JRRM computational performance can be further improved, in particular for high cell loads, using advanced and optimized linear programming software and algorithms. The application of suboptimal JRRM policies has also been shown to significantly reduce the technique's execution time, with only a slight decrease in the achieved user perceived QoS levels for the less priorised services.

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Industry, Tourism and Trade under the project TSI-020400-2008-113 (CELTIC proposal CP5-013), and the Ministry of Science and Innovation under the project TEC2008-06728.

REFERENCES

[1] 3GPP, *Improvement of Radio Resource Management (RRM) across RNS and RNS/BSS, TR 25.881, v5.0.0*, Jan. 2002.

[2] 3GPP, *Improvement of Radio Resource Management (RRM) across RNS and RNS/BSS post-Rel-5*, TR 25.891, v0.3.0, June 2003.

[3] J. Perez-Romero, O. Sallent, R. Agusti, "Policy-based Initial RAT Selection Algorithms in Heterogeneous Networks", in *Proc. of 7th IFIP International Conference on Mobile and Wireless Communication Networks (MWCN'05)*, Marrakech (Marocco), Sept. 2005.

[4] S.J. Lincke, "Vertical Handover Policies for Common Radio Resource Management", *International Journal of Communication Systems*, vol. 18, no. 6, pp. 527–543, Aug. 2005.

[5] A. Tolli, P. Hakalin, H. Holma, "Performance Evaluation of Common Radio Resource Management (CRRM)", in *Proc. IEEE International Conference on Communications (ICC 2002)*, vol. 5, pp. 3429-3433, New York (USA), April 2002.

[6] O. Cabral, F.J. Velez, J. Rodriguez, V. Monteiro, A. Gameiro, N.R. Prasad, "Optimal Load Suitability Based RAT Selection for HSDPA and IEEE 802.11e", in *Proc. 1st International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE 2009)*, pp.722-726, Aalborg (Denmark), May 2009.

[7] Y. Guang, C. Jie, Y. Kai, Z. Ping, V.O.K. Li, "Joint Radio Resource Management based on the Species Competition Model", in *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2006)*, pp. 54-57, Las Vegas, Nevada (USA), April 2006.

[8] O.E. Falowo, H.A. Chan, "Joint Call Admission Control Algorithm for Fair Radio Resource Allocation in Heterogeneous Wireless Networks Supporting Heterogeneous Mobile Terminals", in *Proc. 7th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1-5, Las Vegas (USA), Jan. 2010.

[9] L. Giupponi, R. Agusti, J. Pérez-Romero, O. Sallent, "A Novel Approach for Joint Radio Resource Management based on Fuzzy Neural Methodology", *IEEE Trans. on Vehicular Technology*, vol. 57, pp. 1789-1805, May 2008.

[10] M.C. Lucas-Estañ, J. Gozálvez, J. Sánchez-Soriano, "Common Radio Resource Management Policy for Multimedia Traffic in Beyond 3G Heterogeneous Wireless Systems", in *Proc. 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1-5, Cannes (France), Sept. 2008.

[11] J. Gozálvez, M.C. Lucas-Estañ, J. Sánchez-Soriano, "Joint Radio Resource Management in Beyond 3G Heterogeneous Wireless Systems", in *Proc. 11th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1-5, Lapland (Finland), Sept. 2008.

[12] E.A. Yavuz, V.C.M. Leung, "Computationally Efficient Method to Evaluate the Performance of Guard-Channel-Based Call Admission Control in Cellular Networks", *IEEE Trans. on Vehicular Technology*, vol. 55, no. 4, pp.1412-1424, July 2006.

[13] 3GPP, *Services and service capabilities*, 3GPP TS 22.105, version 6.3.0, 2005.

[14] P. Barford, M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *Proc. Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE 1998)*, pp. 151-160, June 1998.

[15] J. Ho, Y. Zhu, S. Madhavapeddy, "Throughput and Buffer Analysis for GSM General Packet Radio Service (GPRS)," in *Proc. IEEE Wireless Comms. and Networking Conf. (WCNC 1999)*, pp. 1427-1431, Sept. 1999.

[16] O. Lazaro, D. Girma, J. Dunlop, "H.263 Video Traffic Modelling for Low Bit Rate Wireless Communications", in *Proc. of the IEEE Personal, Indoor and Mobile Radio Communications Symposium* (*PIMRC*), vol. 3, pp. 2124-2128, Barcelona (Spain), Sept. 2005.

[17] L. Hanzo, P. Cherriman, J. Streit, *Wireless video communications: second to third generation systems and beyond*, IEEE Press, 2001.

[18] ETSI, *Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Overall Description of the GPRS Radio Interface; Stage 2*, 3GPP TS 03.64, version 8.10.0, 1999.

[19] *3GPP, Technical Specification Group Radio Access Network; Physical layer procedures (FDD)*, 3GPP TS 22.214, version 7.3.0, 2006.

[20] J.F. Nash, "The Bargaining Problem", *Econometrica,* vol. 18, no. 2, pp. 155-162, April 1950.

[21] 3GPP, *Packet-switched handover for GERAN A/Gb mode*, 3GPP TS 43.129, version 6.13.0, 2009.

[22] 3GPP, *Radio subsystem synchronization*, 3GPP TS 45.010, version 6.7.0, 2008.

[23] O. Sallent, "A Perspective on Radio Resource Management in B3G", in *Proc. of the 3rd International Symposium on Wireless Communication Systems (ISWCS'2006)*, pp. 30-34, Valencia (Spain), Sept. 2006.

[24] A.P. da Silva, F.R.P. Cavalcanti, R.A. de O. Neto, "VoIP Capacity Analysis of Wireless Multi-Access Networks using Access Selection Schemes", in *Proc. of the IEEE Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, pp. 1-5, Athens (Greece), Sept. 2007.

[25] MS320TCI100Q DSP Delivers the Industry's Fastest Performance for 3G Wireless Base Stations, Technology for Innovators, 2004 Texas Instruments Incorporated SPRL086B.

[26] TMS320C6455 Data Sheet, Fixed-Point Digital Signal Processor, Texas Instruments, SPRS276I, May 2005, revised April 2009.

[27] Code Composer Studio (CCStudio) Integrated Development Environment (IDE) v4.x, URL: http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html.

[28] F. S. Hillier, G. J. Lieberman, *Introduction to operations research*, 7th edition, McGraw-Hill, 2001.

[29] ILOG CPLEX, URL: http://www.ilog.com/products/cplex/index.cfm.

[30] LP_Solve, URL: http://lpsolve.sourceforge.net/5.5/.

[31] F. Brglez, J.A. Osborne, "Performance Testing of Combinatorial Solvers with Isomorph Class Instances", in *Proc. 2007 Workshop on Experimental Computer Science*, San Diego (California), June 2007.

[32] S. Bayliss, C.-S. Bouganis, G.A. Constantinides, W. Luk, "An FPGA Implementation of the Simplex Algorithm", in *Proc. 2006 IEEE International Conference on Field Programmable Technology (FPT 2006)*, pp. 49-56, Bangkok (Thailand), Dec. 2006.

[33] M. Carlsson, G. Ottosson, "A Comparison of CP, IP and Hybrids for Configuration Problems", Swedish Institute of Computer Science, Technical Report T99:04, 1999.

[34] H. Mittelmann, Benchmarks for Optimization Software, Arizona State University. Available: http://plato.asu.edu/bench.html.

[35] Harri Holma, Antti Toskala, *HSDPA/HSUPA for UMTS. High Speed Radio Access for Mobile Communications*, John Wiley & Sons Ltd, 2006.

M.C. Lucas-Estañ received a Telecommunications Engineering degree in 2007 and a DEA in industrial and telecommunications technologies in 2009, both from the University Miguel Hernandez of Elche in Spain. In 2007, she joined the Uwicore research laboratory where she is conducting her PhD on joint radio resource management policies in multimedia heterogeneous wireless systems. She received an Orange Foundation award at the 2007 Spanish Young Scientist Contest organized by the Spanish Ministry of Education for her research on Radio Resource Management for wireless networks. She is currently contributing to the ICARUS Celtic and mHop projects.

J. Gozalvez received an electronics engineering degree from the Engineering School ENSEIRB (Bordeaux, France), and a PhD in mobile communications from the University of Strathclyde, Glasgow, U.K. Since October 2002, he is with the University Miguel Hernandez of Elche, Spain, where he is currently an Associate Professor and Director of the Uwicore Laboratory. At Uwicore, he is leading research activities in the areas of wireless vehicular communications, radio resource management, heterogeneous wireless systems, and wireless system design and optimization. He currently serves as Mobile Radio Senior Editor of IEEE Vehicular Technology Magazine, and previously served as AE of IEEE Communication Letters. He is TPC Co-Chair of the 2011 IEEE VTC-Fall, and was TPC Co-Chair of the 2009 IEEE VTC-Spring, and General Co-Chair of the 3rd ISWCS 2006. He is also the founder and General Co-Chair of the IEEE International Symposium on Wireless Vehicular communications (WiVeC) in its 2007, 2008, and 2010 editions. He has been elected to the Board of Governors of the IEEE Vehicular Technology Society (2011–2013) and to the IEEE Distinguished Lecturers program of the IEEE Vehicular Technology Society.