COMPUTING AND RADIO RESOURCE MANAGEMENT INTERACTIONS IN FLEXIBLE RADIO ENVIRONMENTS

Jose Salazar Ismael Gomez Vuk Marojevic Antoni Gelonch
Polytechnic University of Catalonia, Barcelona, Spain

ABSTRACT

Due to the wireless industry tendencies, different radio access technologies will coexist in a heterogeneous environment. Software Defined Radio, as a concept that tries to provide flexibility, might ease the integration of that environment. The goal of this work is to introduce the concept of computing resource management, that working in cooperation with the radio resource management strategies leading to minimize computing costs assuring the same or better QoS in such future heterogeneous systems. Furthermore, the action of computing management might help to bring balance over the communication load.

I. INTRODUCTION

Future wireless communications systems characterized by a mix of heterogeneous radio access technologies (RAT's) rapidly evolving seems to push for hardware and software technologies capable to increase the level of flexibility of the existing wireless infrastructures and terminals. In addition some of the main characteristics of the new wireless environment is the high increase in computational demand of the new wireless standards and the need to roam from one Radio Access Technology to another while maintaining the current session. Additional flexibility is required to handle the variety of multimedia services and to provide adaptability to the changing system requirements in order to handle diverse data types and operational time-variant environments. It is clearly stated that the radio processing chain of the wireless systems changes dramatically depending of the service going to be provided and the current conditions in the scenario.

Although several degrees of flexibility can be accomplished by different approaches, full flexibility can only be assured by assuming the Software Defined Radio (SDR) concept which intends to cope with several mobile communication standards by implementing them by means of software. Therefore SDR enables implementation of different air interface standards as software modules and multiple instances of such modules can co-exist in infrastructure equipment and mobile units [1, 2].

In the first instance, referring to the radio issues, the Common Radio Resource Management (CRRM) represents a concept which tries to manage dynamically the allocation and de-allocation of radio resources (e.g. time slots, codes, frequency carriers, etc.) assuring the required QoS the service provided in an heterogeneous access environment [11, 12, 13].

On the other hand, it is noticeable that computing resource management in the future SDR terminals and Base Stations is of prime importance for the success on its implementation [2]. The user terminal and also the Base Stations will have to present enough resources in terms of memory and computing power, to meet the user needs. At difference of the current actuation, where over provisioning is assumed, the resource management will have to know what are the required resources to sustain the requested service as well as what does the terminal hardware platform can provide to make its implementation possible. Also, mobile units have one more restrictive constraint: power consumption. Thus, a more efficient approach requires a tough coordination of the usage of computing resources with the radio ones pushing for a stronger interaction between the SDR platform and the Radio Resources Management entities.

II. RESOURCE MANAGEMENT

A. Computing Resource Management

The management of computing resources, which refer to the hardware that implements the (software-defined) signal processing chains for radio communications, lead into a framework suitable for trading off communication against computation, i.e. an intelligent wireless communication system seeking an efficient usage of both computing and radio resources. However, the final goal of the Computing Resource Management entity (CRM) is to minimize the amount of spent resources to perform the processing tasks of the radio processing chain and adapting them to the changing environment while maintaining the assumed user QoS. Thus we should take into account several parameters to express the requirements of the functions to be accomplished making emphasis in one of them, the computing costs, expressed in million of operations per second (MOPS). Savings can be realized for example, through reducing the arithmetic complexity by decreasing the number of arithmetic operations during the computationally most intensive operations involved [6] or by changing the current algorithm for another, with different cost and performance, more adapted to the scenario.

B. Computing and Radio interactions

It must be noticed that when the Computing Resource Management entity (CRM), as in Fig. 1, handles the reconfiguration due to a response to the surroundings, it might seeks a balance between QoS and minimal computing consumption. It ought to interact with the Radio Resource Management unit (likewise Common-RRM) taking information about the available radio resources and the strategies that can be completed to satisfy the quality parameters [11]. On the other hand it also needs to cooperate with the handset, so that it can identify its available computing capacity, architecture, remaining power etc.

III. CASE STUDY

It is well known that a WCDMA system, like any other communication system, has a limited capacity. The number of users to whose is capable to give service is restricted by the transmission rate of each user, the propagation losses due to the distance between the mobile unit and the base station, and the network load, and the level of interference.

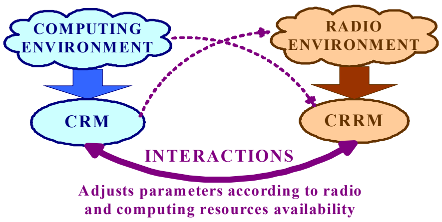

Figure 1: Interaction between the Computing Resource Management and the CRRM

The RRM strategies lead with the problem indicated above, preserving a certain quality target for each user in term of the Bit Energy to Noise Ratio (E_b/N_0). It is also well known that some modules in the processing chain can improve the resulting QoS by increasing their computing costs. This idea aims for a joint interaction of the Computing Cost (CRM) and Radio Resource Management (CRRM) that shall bring a more robust performance within the system by reducing computing costs while maintaining QoS.

A. Resources requirements of RATs Processing Chains

The Fig. 2 shows, as example, the task graph equivalent to the digital signal processing chain of the physical layer of a software-defined UMTS user up/downlink receiver [7]. In such figure has been differentiated the sections corresponding to the Chip-Rate processing and the Bit-Rate processing and the corresponding resources requirements particularized for a 128 kbps service. These computing requirements are estimates from available implementations [9, 10, and 11]. It is important to remember that an instantaneous BER constraint implies that a user can transmit on the channel only if his instantaneous (E_b/N_0) is above a specified target level. The output performance of several algorithms in the chain, and consequently its computing requirements, depends of the quality of the service demanded and the performance of the data at the input of the each one of the modules [6, 7, 8]. One of the most relevant is the Turbo Decoder which will be analyzed in more detail, as an example, to show the importance of the relationship between CRRM and CRM. The complete figure has been included to give a flavor of the importance of the Turbo Decoder cost and performance.

B. Turbo Decoder Performance

Turbo decoders are of iterative type; the decoder works on a received frame iteratively (remember that UMTS frames have length values between 40 and 20730 bits [7]) so that the error on the frame decrease with the Number Of Iterations (NOI). It is for sure that turbo code deliver excellent BER and SNR performance assuring a relative good performance with bad signal conditions (for example BER<10^{-3} for E_b/N_0 of 1 dB) [3, 4, 5, 6]; nevertheless its achievements come at the cost of intense computing requirements. For instance from Fig. 2 can be denoted that the turbo decoder represents the 5 % of the total computing costs over the whole chain, nevertheless this value is obtained for a 3 iterations implementation.

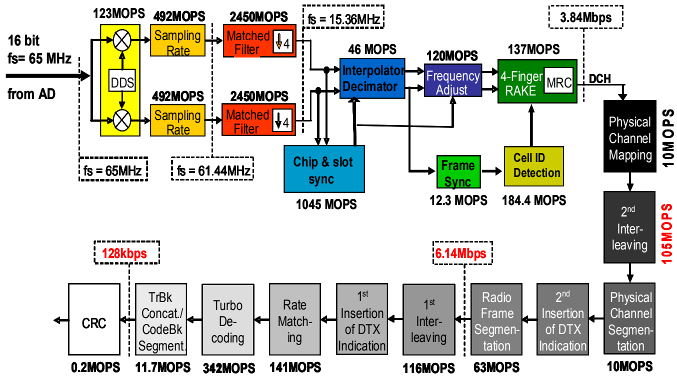

Figure 2: Receiver chain of an UMTS DCH channel for a 128 kbps service.

Fig. 3 depicts the BER performance of the turbo decoder for different NOI values, assuming a code length of 1024 bits. It is important to state that the turbo decoder algorithm used to correct the frames is the Max-Log-MAP type. On this graph the following observation is noticed: when the system operates at normal conditions, i.e. with an SNR per bit greater than 2.5 dB, the probability that the turbo decoder makes a mistake will tend to cero.

C. Computing Costs of the Turbo Decoder

The authors in [4] and [6] found out the arithmetic costs per iteration of the turbo decoder, though the total amount of operations per second consumed by a turbo decoder depends also from the transmission rate R, and the hardware implementation. For example, on [4] a digital signal processor TMS320C6416 (such DSP can accomplish 8 operations per cycle) is used as platform, also they suppose the following implementation specifications: a) Max-Log-MAP algorithm assumed to minimize computational costs, b) the DSP has a quantization level metric of 10 bits, but being processed at 32 bits. The computing costs of this processor performing the turbo decoder algorithm enclose a mean of 114 cycles/bit for each iteration, thus, for a 128 kbps service entails 116.6 MOPS. Moreover at 2 Mbps data rate and 20 iterations it will consume about 2.5 GOPS.

Since the total number of operations per bit consumed by the turbo decoder increases with the number of iterations, designers need to adjust the parameters in order to meet stringent UMTS transceiver requirements and to reduce computing resources. In relation to this problem, some turbo decoding stopping criteria have been proposed based on the denominated dynamic halt condition, which is a halting criteria based on the log-likelihood ratio (LLR) [3, 4].

$$\begin{aligned} &C_C(i) = C_C * N_R * N_{op} \quad ops \\ &C_C = \text{arithmetic costs per iteration} \\ &N_R = 128\text{ k for } R = 128\ kbps\ \ \&\ \ TTI = 10\ ms \\ &N_{op} = \text{Number of Operations per cycle} \end{aligned} \tag{1}$$

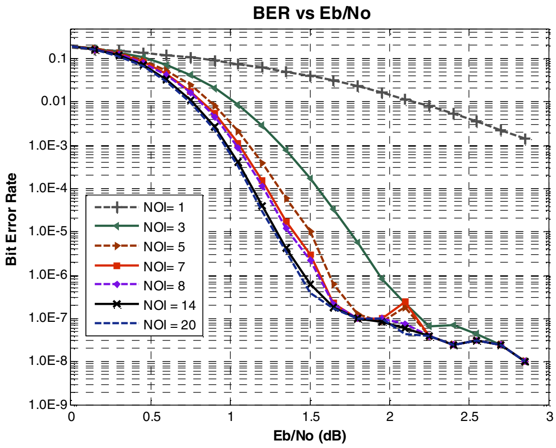

Figure 3: Performance Behaviour of the Turbo decoder

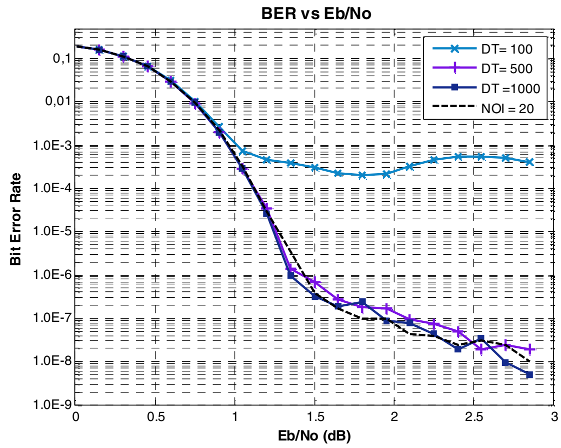

Figure 4: Performance Behaviour of the Turbo decoder for different ΛT values.

D. *The dynamic halt condition*

The decoder estimate a probability for all the bits in the block via the calculation of the LLR's; however, here the decoder stops once the absolute value of all the LLR are above a threshold ΛT, avoiding an unnecessary iteration:

$$\begin{aligned}&\min_{1\le k\ll C_L} = \{|\Lambda_2(X_k)|\} > \Lambda_T\\&C_L = \text{code length}\\&\Lambda_2(X_k) = LLR\\&\Lambda_T = \text{dynamic halt threshold}\end{aligned} \qquad (2)$$

It comes to reason that, when dynamic halt is used, the BER parameter is a function of the input E_b/N_0. This is illustrated on Fig. 4. Conversely, the performance in terms of computational requirements of the algorithm is highly dependent on the choice of Λ_T. Fig. 5 illustrate how for different Λ_T, the number of iterations to stop the turbo decoder, grows as the Λ_T increases. Notice that the quantization process used during the implementation of the turbo decoder is as in [4]. This explains why the results for the Λ_T, are of the order of 1000 instead of 10 as the results found in [3].

IV. Case Study Results

The purpose of the study is to control the number of iterations on the turbo decoder, in order to exchange the computing consumption and radio resources, taking into account diverse values of the E_b/N_0, and the quality constraints defined for user service. Assuming a user with a data service at 128 kbps, and a suitable BER of 10^{-3}, and according to Fig. 4 and Fig. 5, the following can be assured: the E_b/N_0 required to achieve the desired BER is close to 1 dB, for a Λ_T equal to 1000 the turbo decoder must accomplish at least 10 iterations, which means high computing costs per user.

However, it can be seen that the turbo decoder can reach the desired BER using a Λ_T of 100 with just 4 iterations. A similar analysis can be completed for several BER values. Next, it can be outlined how the system may adapt the parameters.

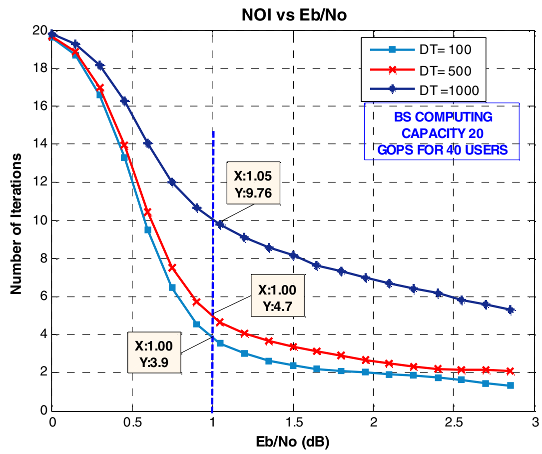

Figure 5: Average number of iterations required for different Λ_T values

A. *Computing Costs and E_b/N_0*

Considering the uplink in a UMTS radio defined environment trade-offs between computing costs, and radio resources at the Base Station can be achieved. The CRRM interacts with the Computing Management in order to exchange computing and radio resources. As consequence, the radio scenario conditions and the algorithm assumed in the RRM/CRRM entities produced a distribution of target E_b/N_0 for assuring certain level of QoS to the considered user.

The wireless environment will modify such distribution therefore the E_b/N_0 distribution received at the BS side will be slightly different from the target one, as illustrated in Fig. 6.

In addition, the CRM will handle the turbo decoder computing costs, in this case, selecting a Δ_T value suitable for the conditions. The related results of computing costs for the Turbo Decoder, that are calculated using (1), for dynamic halt of 100, 500 and 1000.

- Λ_T = 100: for this value, the turbo decoder algorithm halts by completing a mean of iterations that brings out a small amount of computing costs (average of 305.9 MOPS), from this result then, 100 is a suitable choice when the desired BER is 10^{-3}.

- Λ_T = 500: for this case, the turbo decoder carries out a relative little mean of 3.3 iterations to stop. Hence, the mean of computing costs remains also small (383 MOPS).
- Λ_T = 1000: here, it is obtained that the turbo decoder experience a mean NOI of 6.15 with an associate mean of computing costs of 718 MOPS.

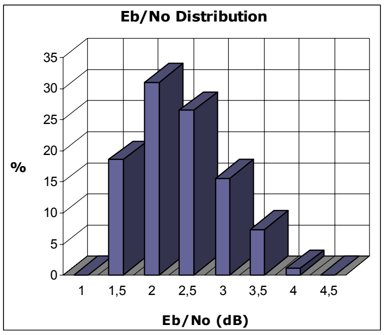

Figure 6: E_b/N_0 received at the Base Station.

Then, it is significant to calculate the mean of weighted computing costs (relative to the E_b/N_0 distribution). The mean of computing costs (CC_{Mean}) for different Λ_T values come in Table 1.

Table 1: Mean Computing Costs for different dynamic halt threshold related to the E_b/N_0 distribution.

Λ_T	CC_{Mean}
100	2.0547 GOPS
500	4.5999 GOPS
1000	9.8348 GOPS

Using these results the normalized percentage of the computing costs to the matching mean costs for each case are obtained, and the related distributions are shown in Fig. 7.

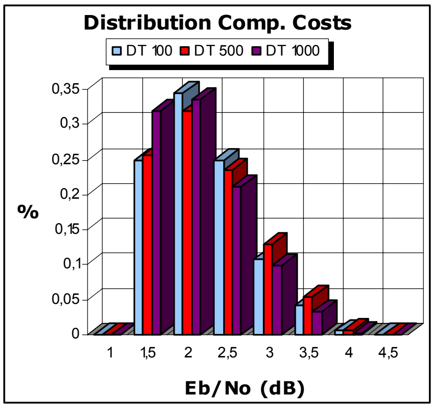

Figure 7: Distribution of Computing Costs in MOPS for different Λ_T.

As it can be observed, the behaviors of the relative computing costs to the E_b/N_0 distribution present a very minor difference when is considered a Λ_T of 100; the same takes place for Λ_T equal to 500, with just a little more concentration of computing costs around the maximum value from the considered E_b/N_0 distribution; but for Λ_T of 1000, the computing costs increase because of the influence of the radio resources, this is visible near the maximum E_b/N_0 at 1.5 dB, 2 dB.

B. *Radio and Computation Capacity*

In a real scenario, the Base Station should assume a certain radio capacity in terms of the number of users using a defined set of services with its associated QoS requirements. Also such Base Station shall be designed to support such number of users in terms of computing resources availability. From previous discussion, it is clear that here emerge important interactions between radio and computing capacities, especially if it is assumed that a dynamic adjustment between the computing resources spends and the real radio conditions could be accomplished. Thus, must be defined how the Energy bit to Noise ratio influences the system capacity.

The expression in (3) point out the maximum number of users of a certain Base Station, and this result must be assumed as the radio constraints for such BS. It shows that for relative big values of E_b/N_0 the system capacity will suffer [12].

$$\sum_{i=1}^{N} \frac{1}{\left(1+\dfrac{W}{\left(\dfrac{E_b}{N_0}\right)_i R_{b,i}}\right)} < 1 \qquad (3)$$

N = number of users; $R_{b,i}$ = data rate;

W = bandwidth

On the other hand, regarding the computing part it could be assumed that the BS computing resources are enough for 35 users, leaving a total of 15 GOPS for the Turbo decoder (around 4 NOIs in average).

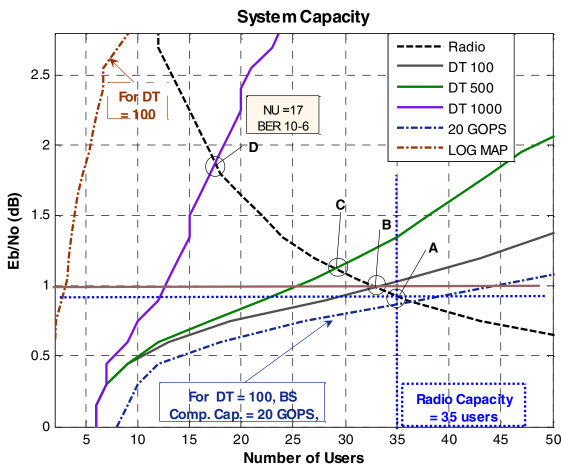

Figure 8: Behaviour of the System Capacity.

On Fig. 8, the next aspects are illustrated: a) the minimum E_b/N_0 required for assuring the 35 users in the radio system is around 0.95 dB (point A), which could not be supported by the current implementation of the Turbo decoder if the requested BER is 10^{-3} (remember that the minimum E_b/N_0 requirement for the Turbo decoder is 1 dB) ; b) the closest point to the previous assumption, taking into consideration the computing resources availability in the BS, is achieved by using a Λ_T =100, assuring a BER of 10^{-3} but only for 33 users (point B); c) If we assume a Λ_T =500, then the number of users is reduced to 28, but a greater BER 10^{-5} could be achieved (point C); d) but if we assume a value of the Λ_T =1000, the maximum number of users that can be served due to computational resources limitations is reduced to 17 (point D) although as it is achieved in terms or radio resources at higher E_b/N_0 due to the Turbo decoder behavior the resulting BER is around 10^{-6}; e) If the computing capacity that is leaved to the Turbo decoder is increased to 20 GOPS, then with Λ_T =100, the target will be out of bound in relation to the system capacity.

Therefore if the service needs some E_b/N_0 minimum to achieve a certain BER (for instance, above 1.5 dB for BER 10^{-6}) the system may well decide to dispatch some radio resources, and modify the computing resources, thus the parameter Λ_T should be set to 1000. This last statement will also assure that the maximum number of users, to which a BS can give service, is sustained; therefore some interchange of resources between the radio and the computing side can be achieved.

C. *Turbo Decoder Implementation Performance*

Following up, will stand a comparison between the previous results and the theoretical results obtained if a Log-MAP algorithm it is used instead of a Max-Log implementation. It is known that from the performance point of view, the turbo decoder will present an improvement of 0.4 dB respects to the Max-Log-Map [3]; even though the Max-Log-MAP algorithm takes about two point three more iterations to converge for a small code, it is a faster algorithm, and more-efficient in terms of computing resources, because the same processor need more processing power to execute an iteration. The Mean computational cost difference between the Max-Log-MAP and the Log-MAP algorithms is 7.17. This also represents a great impact on the system capacity in a limited computational resources BS. This is illustrated also in Fig. 8.

V. CONCLUSIONS

In this work has been observed how for some services and data rates, an UMTS block chain may consume great computing resources. The turbo decoder, which has been deeply analysed, achieves a probability of making a mistake near to cero, though at the cost of intense computing requirements. Therefore, it comes to reason the need of a computing resource management engine, which not only supposes to control the computing resources but to act together with the radio resource management. Trade-offs between radio and computing resources should be accomplished depending on the service requirements, and the availability of both varieties of resources.

It has been exposed that for a certain E_b/N_0 to achieve a BER target a value of the parameter that handle the computational load (in this case the dynamic halt) which is more suitable for both computational and radio resources. Nevertheless as they are also related to the system capacity, the system should decide to raise the E_b/N_0 and to raise the parameter in question - that mean an increase of the computing costs - in order to preserve a certain capacity of users.

Considering implementation aspects in has been demonstrated that although the Max-Log-MAP algorithm used on the implementation requires more decoder iterations than the Log-MAP (theoretically most efficient in performance) at any particular value of E_b/N_0, it needs less than one seventh of the processing power of the log-MAP, which signifies an improvement in computing capacity that has an utter-most impact reflected in system capacity.

The key role of the approach here illustrated is the possibility to adapt the system to different conditions of the wireless communication link, but emphasizing the need of take into account the computing costs.

The next steps are: a) to extend this approach to other blocks on the communication chain; b) to analyze a dynamic control of the parameters, i.e. the computing resource management and the CRRM convey its interactions in a cognitive engine.

ACKNOWLEDGEMENTS

This work has been supported by CYCIT (Spanish National Science Council) under grants TIC2003-08609 and TEC2006-09109, which are partially financed from the European Community through the FEDER program, the DURSI of the Catalonian Government, and European social funds.

REFERENCES

[1] J. Mitola. "The Software Radio Architecture". *IEEE Comm. Mag.*, May 1995.

[2] Lin, Yuan et al., "A system solution for High-Performance, Low Power SDR". *SDR Tech. Conf. Prod. Exp.*, California, November 2005

[3] M. C. Valenti and J. Sun. "The UMTS Turbo Code and an Efficient Decoder Implementation Suitable for Software-Defined Radios". *Int. Jour. Wireless Inf. Net.*, Vol. 8, No. 4, Oct. 2001

[4] Josep Codina M. and Ismael Gómez M. "UMTS/WLAN reprogrammable devices: bit flow processing". *Thesis submitted to award the grade of Telecomm. Eng. Esc. Sup. Castelldefels UPC*, Spain, Feb. 2005.

[5] P. Robertson, E. Villebrun, and P. Hoeher. "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain". *Proc. Int. Conf. Comm.*, p. 1009–1013, June 1995

[6] L. T. Smit et al., "Influences of rake receiver/turbo decoder parameters on energy consumption and quality". *Proc. Int. Conf. Third Generation Wireless and Beyond*, p. 227–235, May 2002.

[7] 3GPP; "Technical Specification Group Radio Access Network. 3G" *TS 25.212 v6.2.0: Multiplexing and channel coding (FDD) (Release 6),* Available: http://www.3gpp.org

[8] Y. Hawwar et al., "3G UMTS Wireless System Physical Layer: Baseband Processing Hardware Implementation Perspective". *IEEE Comm. Mag.*, Sep. 2006.

[9] R. Tanner, J. Woodard, WCDMA – Requirements and Practical Design. *John Wiley and Sons Inc*., 2004.

[10] T. Faber, M. Schönle, "DSP-platform target report," *SLATS Consortium, Project no. 27016, Deliverable D23*, Dec. 1999.

[11] R. Ferrús et al., "A feasible approach for E2E QoS management in coordinated heterogeneous radio access networks". *IEEE Int. Perf. Comp. Comm. Conf.*, . April, 2005.

[12] J. Perez-Romero, R.Agustí, O. Sallent,M. Díaz-Guerra,. "Radio Resource Management strategies in UMTS". *John Wiley and Son Ltd*., 2005, England.

[13] 3GPP; "Technical Specification Group Radio Access Network. 3G" TR 25.881 v5.0.0: "Improvement of Radio Resource Management (RRM) across RNS and RNS/BSS" (Release5), Available: http://www.3gpp.org