An Efficient Dynamic Resource Allocation Algorithm for Packet – Switched Communication Networks Based on Hopfield Neural Excitation Method[1]

D. Calabuig, J. F. Monserrat, D. Gómez-Barquero and O. Lázaro

Abstract: Dynamic Resource Allocation (DRA) algorithms set up different connections over the same resources and perform a scheduling policy to distribute the resources usage. Recently, intelligent DRA techniques based on Hopfield Neural Networks computational methods have been proposed, showing their potential for solving this kind of complex optimization problems. However, the initial algorithms suffer from severe instability problems impacting performance. This paper addresses these specific limitations stressing the proper neuron dynamics and proposing an efficient energy formulation and an optimum calculation of the weighting coefficients. These changes result in a maximum resource utilization together with an optimized neural network convergence.

I. INTRODUCTION

Dynamic Resource Allocation (DRA) algorithms have been widely used to fairly distribute the scarce radio resources of mobile and wireless systems. The main objectives of the DRA algorithms are to maximize the system capacity (bandwidth utilization) and to fairly distribute the resources among the different connections, while satisfying users Quality of Service (QoS). Such algorithms

[1] NOTICE: this is the author's version of a work that was accepted for publication in Neurocomputing. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Neurocomputing, vol. 71, no. 16 – 18, pp. 3439 – 3446, 2008, DOI:10.1016/j.neucom.2007.10.009.

require a multi-objective optimization problem to be solved, which entails an enormous search space (the DRA is a NP-complete optimization problem). Besides, the DRA algorithm has to operate in real time, which complicates obtaining sufficiently good solutions. Most of the techniques reported so far, see e.g. [1]-[4], are either incapable of achieving optimum resource allocation or cannot operate in real time.

Hopfield Neural Networks (HNN) are fast and parallel combinatorial optimizers which have been recently proposed as an effective real time solution to fairly distribute and maximize the resources usage [5]. This seminal work by Ahn and Ramakrishna has further inspired the recent application of HNN-DRA schemes within UMTS [6]. However, the HNN-DRA scheme proposed in [5] suffers from important limitations, which may carry an underestimation of the actual potential of HNN for solving the DRA optimization problem. Firstly, the proposed neurons dynamics formulation is unable to guarantee the system convergence towards the energy minimum what raises a reasonable doubt about the validity of this solution. Secondly, with the current formulation, the users whose individual bandwidth requirement contributes to an aggregated demand that exceeds the overall system resources available, instead of being individually penalized, affect the complete system operation. This fact, together with an improper weighting coefficients calculation, leads to unstable non-optimal solutions which reduces the overall system performance. The aim of this paper is hence twofold: on the one hand, to present the solution to the aforementioned HNN dynamics handicaps of [5], and on the other hand, to demonstrate the feasibility of optimum user-centric resource allocation through an enhanced HNN formulation and a proper weighting coefficient calculation.

II. HNN DYNAMICS

Hopfield Neural Networks are a type of recurrent NN completely characterized by an energy function E that describes their dynamics (i.e., the time evolution of the HNN). See the main contribution of Hopfield in [7] and [8]. HNNs have been widely used in various scientific domains [5],[6],[9] providing feasible solutions to very complex optimization problems within a very short time. From the original formulation of the energy function in [7], Abe [9] obtained the following HNN dynamics equation:

$$\frac{dU_i}{dt} = -\frac{1}{C_i}\frac{\partial E}{\partial V_i} \qquad (1)$$

where U_i is the input tension to the *i-th* neuron, V_i is the corresponding output, both related by the gain function $V_i = g_i(U_i)$, and C_i is the input capacitance. This equation implies that stable solutions (i.e., $dU_i/dt = 0$), occur at energy minima.

Despite the most commonly used gain function is the sigmoid with a finite shape parameter which entails (1), an infinite shape parameter can be assumed [8]. This approximation simplifies the identification of the hardware elements (i.e., the interconnection matrix and the bias) but changes the HNN dynamics equation. Ahn assumed an infinite shape parameter what led him to the following HNN dynamics equation [5]:

$$\frac{dU_i}{dt} = -\frac{1}{C_i}\frac{\partial E}{\partial V_i} - \frac{U_i}{C_i R_i} \qquad (2)$$

where R_i is the equivalent input resistance.

As it can be observed, in case U_i is not equal to zero HNN stable states do not correspond with the energy minima (if $dU_i/dt = 0$ then $\partial E/\partial V_i = -U_i/R_i \neq 0$).

This fact is critical, since the energy function is designed to be minimal for the desirable solutions. Only if the shape parameter is infinite then $U_i \approx 0$, and therefore the system reaches the energy minimum. However, the consideration of an infinite shape parameter makes neurons to take only two values, i.e. the maximum and minimum, involving a non-continuous neuron evolution and larger oscillations. The oscillations are provoked by the fact that an abrupt change in a neuron output implies an abrupt change in the network behavior. Surely due to this fact, a shape parameter of 1 is applied in [5] to suppress the oscillations at the neuron outputs caused by an infinite one, which does not suffice to make the simplification proposed by Hopfield in [8] pertinent. From this simple mathematical analysis, two important conclusions can be drawn. First, the system described in [5] is unable to reach the energy minimum, at least with the chosen parameters. Second, a finite shape parameter should be employed in order to avoid system oscillations, being therefore necessary the usage of the HNN dynamics formulation expressed in (1).

III. EXISTING HNN-BASED DRA ALGORITHM

At the essence of a HNN is an energy function ensuring that global minimum occurs in optimum solution [8]. This feature has been extensively used to accommodate the specification and solution of complex optimization problems. In [5] Ahn is assisted by a HNN to solve the DRA problem to fairly distribute the total system bandwidth among users while maximizing the bandwidth utilization. To this effect, a 2D HNN (with *N*x*M* neurons, *N* users and *M* bit rates) is presented. Neuron outputs represent the bit rates allocated to each user, being

$V_{ij}=1$ if the j-th bit rate is assigned to the i-th user and $V_{ij}=0$ otherwise. The energy function proposed is as follows [5]:

$$E=\frac{\mu_1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}C_{ij}V_{ij}+\frac{\eta^{\zeta}\mu_2}{2}\left|1-\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{B_{ij}}{B_T}V_{ij}\right|+\frac{\mu_3}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\psi_{ij}V_{ij}$$
$$+\frac{\mu_4}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}V_{ij}\left(1-V_{ij}\right)+\frac{\mu_5}{2}\sum_{i=1}^{N}\left(1-\sum_{j=1}^{M}V_{ij}\right)^2 \quad (3)$$

This energy function is composed by five terms. The first one ensures that a fair resource allocation among connections is obtained. The cost function C_{ij} is defined as [5]:

$$C_{ij}=\frac{\left[B_T\left(\frac{B_{i1}}{\sum_{k=1}^{N}B_{k1}}\right)-B_{ij}\right]^2}{\max\limits_{\substack{\forall x\in\{1,\cdots,N\}\\ \forall y\in\{1,\cdots,M\}}}\left[B_T\left(\frac{B_{x1}}{\sum_{k=1}^{N}B_{k1}}\right)-B_{xy}\right]^2} \quad (4)$$

where B_{i1} is the highest bit rate of user i, B_{ij} is the j-th bit rate of user i and B_T is the maximum system capacity. The cost function takes values in the interval [0,1]. $C_{ij}=0$ if B_{ij} is the fair bit rate, and $C_{ij}=1$ when the distance to the fair bit rate is maximum. The fair bit rate of the i-th user is:

$$\tilde{B}_i=B_T\left(\frac{B_{i1}}{\sum_{k=1}^{N}B_{k1}}\right) \quad (5)$$

And the fairness of the allocation can be measured as [5]:

$$\text{fairness}=1-\frac{1}{N}\sum_{i=1}^{N}\left|1-\frac{B_i}{\tilde{B}_i}\right| \quad (6)$$

where B_i is the bit rate finally allocated to user i.

The second term aims at maximizing the allocated bit rate to each user without exceeding B_T. It is worth noting that through η^{ζ}, where

$\zeta = u\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\left(B_{ij}/B_T\right)V_{ij} - 1\right)$ and η is a constant, a high penalty is imposed if the maximum system capacity is exceeded. The third term prevents particular bit rates from being allocated to specific users, and the matrix ψ represents a service bit rate permission table. The fourth term is intended for reaching an stable solution where $V_{ij} \in \{0,1\}$. Finally, the fifth term guarantees that only one bit rate is allocated to each user among the feasible set.

It should be noted that equation (3) is not a Lyapunov function (i.e. it has not a continuous derivative), and, thus, the results about stability obtained by Hopfield cannot be applied. As it can be observed, two variables depend on V_{ij} in the gradient of the second term, namely ζ and the absolute value. The first one is a unit step function whose derivative is always null except at 0 where it is not defined. This discontinuity does not imply any problem because the step function derivative can be defined as 0 at the discontinuity, as in [5]. However, the derivative of the absolute value is also not defined at 0, but in this case the lateral limits differ and have opposite signs. This way, each time B_T is exceeded the outputs of **all neurons** are reduced being subsequently increased in the next iteration, when the total bit rate is below B_T. Consequently, the system in [5] will oscillate, in spite of defining the value of the derivative at the discontinuity. This oscillation has severe consequences for the HNN-DRA performance since it prevents the network from reaching a stable solution. The formulation in [5] forces the system to change abruptly its evolution (i.e., its gradient), when ζ is activated.

IV. USER BANDWIDTH USAGE-DRIVEN HNN-DRA ALGORITHM (UB-HNN-DRA)

This section presents an enhanced HNN-DRA formulation, called UB-HNN-DRA, which proposes a new second term in (3), so that the HNN dynamics drawbacks identified in the previous section are effectively addressed. The UB-HNN-DRA follows the same objective as the HNN-DRA algorithm. However, for the reasons detailed below, the second term has been redefined as follows:

$$-\frac{\mu_{2a}}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{B_{ij}}{B_T}V_{ij}+\frac{\mu_{2b}}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\xi_{ij}V_{ij} \qquad (7)$$

$$\xi_{ij}=u\left(\frac{H_{ij}}{B_T}-1\right) \text{ and } H_{ij}=B_{ij}+\sum_{\substack{n=1\\n\neq i}}^{N}\sum_{m=1}^{M}B_{nm}V_{nm} \qquad (8)$$

where H_{ij} is the system total bit rate if the *j*-th bit rate is allocated to the *i*-th user and the rest of users are assigned the current bit rate status. Note that the first component, weighted by μ_{2a}, always maximizes the overall allocated resources. The second component, weighted by μ_{2b}, now permits that only the users whose bit rate demands make H_{ij} exceed B_T be penalized for such behavior as opposed to [5], where such penalty affected every user.

The gradient of the new proposed term is:

$$\frac{\partial}{\partial V_{ij}}\left(\frac{\mu_{2b}}{2}\sum_{k=1}^{N}\sum_{l=1}^{M}\xi_{kl}V_{kl}\right)=\frac{\mu_{2b}}{2}\left[\sum_{k=1}^{N}\sum_{l=1}^{M}V_{kl}\frac{\partial \xi_{kl}}{\partial V_{ij}}+\xi_{ij}\right]=\frac{\mu_{2b}}{2}\xi_{ij} \qquad (9)$$

In (9) it is also necessary to calculate the derivative of ξ_{ij} and consequently the derivative of the step function. Therefore, the new energy function is not a Lyapunov function either and a similar discontinuity appears as in (3), but with the new formulation (7), the penalty is isolated to each individual neuron (i.e., it is only applied to the neurons that change their ξ_{ij}). This improvement allows some of the neurons to retain its natural stable evolution and also the overall HNN to reach a stable state where beforehand oscillation was mandatory and optimum solution could not be reached. However, the achieved fairness is higher with the original formulation. Since a resource allocation is fairer when all user bit rates approach the proportional distribution, when employing the original formulation, according to which all users' neurons are jointly reduced, the fairness among them is not broken. On the other hand, with the new formulation, reducing the neuron outputs of one single user benefits the overall system neurons evolution at the expense of decreasing the fairness of that user.

Although stability is not mathematically ensured, this new second term reduces the neurons oscillation and, together with the correct definition of the HNN dynamics (1), both changes have profound implications in the dynamics of the HNN, the stability of the solutions reached and ultimately in the overall system performance.

V. WEIGHTING COEFFICIENTS CALCULATION

The HNN-DRA weights obtained in [5] do not achieve the best performances in terms of throughput maximization and system constraints satisfaction. The throughput is maximized and the constraints are fulfilled when the total allocated bit rate approaches B_T without exceeding it, and ensuring as much as possible the fair distribution of bit rates among users. This section presents a new rationale for the weighting coefficients calculation.

A. HNN-DRA.

The fourth term of the HNN-DRA energy function only aims at enhancing the convergence speed of the neural network. This term must not avoid the change of a neuron output, from 0 to 1, or vice versa, if the rest of the terms point to it. Let (i,a) and (i,b) be two neurons of the same user. In case neither of their correspondent bit rates exceeds the maximum resources, the energy gradient of both neurons is:

$$\frac{\partial E}{\partial V_{ia}} = \frac{\mu_1}{2} C_{ia} - \frac{\mu_2}{2}\frac{B_{ia}}{B_T} + \frac{\mu_4}{2}(1-2V_{ia}) - \mu_5\left(1-\sum_{l=1}^{M} V_{il}\right)$$

$$\frac{\partial E}{\partial V_{ib}} = \frac{\mu_1}{2} C_{ib} - \frac{\mu_2}{2}\frac{B_{ib}}{B_T} + \frac{\mu_4}{2}(1-2V_{ib}) - \mu_5\left(1-\sum_{l=1}^{M} V_{il}\right)$$

Since the energy function is a second order function, if the bit rate B_{ia} is better than B_{ib} in terms of fairness and throughput maximization then:

$$\frac{\mu_1}{2} C_{ia} - \frac{\mu_2}{2}\frac{B_{ia}}{B_T} < \frac{\mu_1}{2} C_{ib} - \frac{\mu_2}{2}\frac{B_{ib}}{B_T} \quad (10)$$

The worst case occurs when $V_{ib}=1$ and $V_{ia}=0$. Then, bearing in mind that a neuron output is further increased if its gradient decreases (see (1) or (2)), the fourth term will not prevent the change to $V_{ib}=0$ and $V_{ia}=1$ if:

$$\frac{\partial E}{\partial V_{ia}} < \frac{\partial E}{\partial V_{ib}}$$

$$\frac{\mu_1}{2} C_{ia} - \frac{\mu_2}{2}\frac{B_{ia}}{B_T} + \frac{\mu_4}{2} < \frac{\mu_1}{2} C_{ib} - \frac{\mu_2}{2}\frac{B_{ib}}{B_T} - \frac{\mu_4}{2}$$

$$\mu_4 < \min\left[-\frac{\mu_1}{2}(C_{ia}-C_{ib}) + \frac{\mu_2}{2}\frac{B_{ia}-B_{ib}}{B_T}\right] \quad (11)$$

(11) gives an upper bound for μ_4. Note that the right side of (11) can never be negative thanks to (10).

Now focusing on the fifth term, users should never have more than one bit rate allocated. The fifth term is minimum when all the neuron outputs of a user sum one. At these points this term and its derivative are zero. If the sum of the derivatives of the rest of terms is not zero, neurons start to increase or decrease their value running the risk of pushing the outputs away from the desired value. Let define δ as the maximum desired distance from the sum one for the outputs, i.e. the equilibrium is reached if $\left|1-\sum_{l=1}^{M} V_{il}\right| < \delta$. If $\sum_{l=1}^{M} V_{il} > 1+\delta$ then the following must be satisfied to decrease the neurons output:

$$\frac{\mu_1}{2} C_{ij} - \frac{\mu_2}{2}\frac{B_{ij}}{B_T} + \frac{\mu_4}{2}\left(1-2V_{ij}\right) - \mu_5\left(1-\sum_{l=1}^{M} V_{il}\right) > 0$$

$$\frac{\mu_1}{2} C_{ij} - \frac{\mu_2}{2}\frac{B_{ij}}{B_T} + \frac{\mu_4}{2}\left(1-2V_{ij}\right) + \mu_5\delta > 0 \qquad (12)$$

And if $\sum_{l=1}^{M} V_{il} < 1-\delta$ then to increase the neurons output:

$$\frac{\mu_1}{2} C_{ij} - \frac{\mu_2}{2}\frac{B_{ij}}{B_T} + \frac{\mu_4}{2}\left(1-2V_{ij}\right) - \mu_5\delta < 0 \qquad (13)$$

(11), (12) and (13) are the conditions that must satisfy μ_1, μ_2, μ_4 and μ_5. Therefore, the first step in the weighting calculation is to give an arbitrary value to one of the four coefficients and afterwards to obtain the rest of weights so that they fulfill these three conditions simultaneously. Next, μ_3 and η can be obtained provided these four weights. In the first case, if a specific bit rate is not allowed to a user, then it must not be allocated to him, or in terms of the neural network, the correspondent neuron output must decrease. For that reason:

$$\frac{\partial E}{\partial V_{ij}} > 0$$

$$\frac{\mu_1}{2}C_{ij} - \frac{\mu_2}{2}\frac{B_{ij}}{B_T} + \frac{\mu_3}{2} + \frac{\mu_4}{2}\left(1 - 2V_{ij}\right) - \mu_5\left(1 - \sum_{l=1}^{M} V_{il}\right) > 0$$

In the worst case $C_{ij} = 0$ and $\sum_{l=1}^{M} V_{il} = V_{ij} = 0$ since $\mu_5 \gg \mu_4$ to prevent the system from having more than one bit rate allocated simultaneously to the same user. Therefore:

$$\mu_3 > \mu_2 \frac{\max_{xy}\left[B_{xy}\right]}{B_T} - \mu_4 + 2\mu_5 \qquad (14)$$

Finally, if the maximum system capacity B_T is exceeded, the neuron outputs must be decreased. Consequently, maintaining the same rationale as before:

$$\frac{\mu_1}{2}C_{ij} + \frac{\eta\mu_2}{2}\frac{B_{ij}}{B_T} + \frac{\mu_4}{2}\left(1 - 2V_{ij}\right) - \mu_5\left(1 - \sum_{l=1}^{M} V_{il}\right) > 0$$

$$\eta > \frac{2\mu_5 - \mu_4}{\mu_2}\frac{B_T}{\min_{xy}\left[B_{xy}\right]} \qquad (15)$$

B. UB-HNN-DC.

HNN-DC and UB-HNN-DC weighting coefficients calculation are equivalent, hence, expressions (11), (12), (13) and (14) are still valid rewriting μ_2 as μ_{2a}. Focusing on μ_{2b} and in order to allocate a bit rate that not exceeds the maximum capacity, at least one of the correspondent neurons must be more favored (increasing faster or decreasing slower) than the neurons exceeding the maximum resources, hence:

$$\frac{\partial E}{\partial V_{i,\mathrm{fav}}} < \frac{\partial E}{\partial V_{i,\mathrm{exc}}}$$

where (i, fav) is the favored neuron and (i, exc) is the neuron whose bit rate exceeds the maximum capacity. Therefore, the following must be fulfilled:

$$\frac{\mu_1}{2}C_{i,\text{fav}} - \frac{\mu_{2a}}{2}\frac{B_{i,\text{fav}}}{B_T} + \frac{\mu_4}{2} < \frac{\mu_1}{2}C_{i,\text{exc}} - \frac{\mu_{2a}}{2}\frac{B_{i,\text{exc}}}{B_T} + \frac{\mu_{2b}}{2} - \frac{\mu_4}{2}$$

$$\mu_{2b} > \max\left[\mu_1\left(C_{i,\text{fav}} - C_{i,\text{exc}}\right) - \mu_{2a}\frac{B_{i,\text{fav}} - B_{i,\text{exc}}}{B_T} - 2\mu_4\right] \tag{16}$$

VI. PERFORMANCE EVALUATION

The performance of the proposed formulation, UB-HNN-DC, has been studied in terms of convergence capability, system bandwidth utilization and fairness. The system is bandwidth limited and the maximum capacity is set to 850 kbps, a common value for WCDMA UMTS deployment [6]. A number of valid bit rates {256, 128, 64, 32, 16 kbps} are available to provide a single service within a range of agreed service levels. Thus, the QoS levels observed by the users are directly related to the bit rates allocated, where higher bit rates imply improved quality perception. The HNN target is to maximize the allocated bandwidth serving each user with a bit rate as high as possible. The simulations are carried out with an increasing number of active users in a single cell. Moreover, to abstract time evaluations from the features of any digital computer employed and the actual implementation of the HNN routine, the concept of iteration has been applied; with an iteration representing a complete computation of the neuron states. The maximum number of iterations for each run of the algorithm is set to 10.000, and the results are computed and averaged for a total number of 10.000 runs for each system load, randomly selecting the HNN initial state for each run.

In order to decouple performance degradation due to wireless phenomena such as path loss, noise or interference from performance degradation due to spurious HNN dynamics behavior, such issues are not taken into consideration.

Three different HNN algorithms are compared: the original HNN-DRA with the weights proposed in [5] and in Section V.A., and the UB-HNN-DRA whose weights are taken from Section V.B. Table I summarizes the value of these weights. The rest of the HNN parameters are extracted from [5] and set to:

$$R_i = 1\ \Omega,\ C_i = 1\ \mathrm{F},\ i = 1 \ldots N$$

Fig. 1 depicts a typical evolution of the neuron outputs for a particular user when an infinite shape parameter is selected, $\alpha_i = \infty$. Although this figure has been obtained using the formulation and the weights of [5], a similar behavior could be observed with the UB-HNN-DRA formulation and even with any energy function defined to solve any other optimization problem. The oscillations produced by the infinite shape parameter make the neuron evolution to be uncontrollable. Therefore, the selection of $\alpha_i = 1$ is preferred.

With $\alpha_i = 1$, the dynamics of the HNN-DRA with the original weights of [5] and the UB-HNN-DRA algorithms are now studied. Fig. 2 shows a typical evolution of the neuron outputs for a particular user. As it can be observed, the formulation in [5] is unable to decide between 32 and 128 kbps. On the other hand, with the proposed formulation, the oscillations disappear and the system is able to reach the equilibrium point allocating 64 kbps to the user. In this case the oscillations result from the absolute value of the energy function and therefore they are different from those observed in Fig. 1.

It can be argued that this evolution is rather seldom, without any real impact on the long-term average system performance. To this effect, the oscillation probability, $P_{oscillation}$ -probability that the system get at the maximum number of iterations without reaching stability-, has been evaluated. As depicted in Fig. 3, HNN-DRA exhibits a significant oscillation probability for both cases, using the weights of [5] and with those obtained in Section V.A. Such probability is zero only for two special cases: with two and six users. A system with two users has enough resources to assign the maximum bit rate to both users, 256 kbps. In the case 6 users are present in the system, such probability is also zero since the fair allocation, 850/6=141 kbps, is higher and close to 128 kbps therefore the HNN-DRA is stable with all the users allocated with 128 kbps. On the contrary, fair allocation for 4 users, 850/4=212, is now lower and close to 256 kbps and hence the HNN-DRA tries to allocate 256 kbps to all users. Since this allocation exceeds B_T, all the neurons outputs are reduced and consequently the absolute value avoids the HNN-DRA to reach a stable state. This behavior explains the high oscillation probability. The case with 8 users is similar but now the fair allocation is lower and close to 128 kbps. On the contrary, UB-HNN-DRA exhibits no oscillation and hence a quick system convergence is achieved.

Another conclusion that could be drawn from Fig. 3 is that weights of Section V.A. imply worse performance than the ones originally proposed in [5], as the oscillation probability is considerably higher. Nevertheless, the total allocated bit rate shown in Fig. 4 leads to the opposite conclusion. This figure has been obtained averaging the total bit rate allocated of all the feasible solutions obtained with the algorithms. Unfeasible solutions are those with none or more than one neuron active or those that exceed the maximum capacity B_T. The worst bandwidth utilization is exhibited by formulation in [5], especially for large number of active users (more than 15 users in our case). The non satisfactory weights selection makes the allocated bit rates tend to the lowest available one (i.e., 16 kbps for each user). However, such behavior improves the stability of HNN-DRA since the maximum bandwidth is not exceeded, what justifies the best behavior of formulation in [5] regarding oscillation probability. On the other hand, the new weights of Section V.A maximize the bandwidth usage what entails a higher oscillation probability. Consequently the reduction in the oscillation probability of the HNN-DRA formulation is at the expense of a lower utilization of the free bandwidth available. Conversely, the UB-HNN-DRA maximizes the bandwidth usage and couples that resource allocation optimization with a faster convergence due to the oscillation avoidance mechanism embedded in the enhanced formulation. Moreover, regarding the feasibility of solutions, the 18.77% of the solutions of the original formulation and weights are unfeasible and the 12.69% for the new weights, whereas the UB-HNN-DRA has no unfeasible solution in any of the runs performed. Unfeasibility is the worst consequence of oscillation. If the neural network oscillates, DRA solutions extracted from the neuron states are uncontrollable and, thus, they can be unfeasible.

Finally, Fig. 5 compares the fairness (6) of the different formulations. Two different approaches have been used to calculate this fairness. The first one, called average instant fairness, calculates the fairness among users for each run and averages the fairness of the 10.000. The second method, called long-term fairness, averages the bit rate allocated to each user in all the performed runs and next obtains the fairness among them. As Fig. 4, this figure is obtained considering only the feasible solutions. Fig. 5.a) shows the average instant fairness. The UB-HNN-DRA formulation is more unfair as explained in section IV, and the HNN-DRA has similar fairness independently of the employed weights. Nevertheless, the long-term fairness shown in Fig. 5.b) changes considerably. Now, the original formulation of [5] is the more unfair whereas the enhanced version with modified weights and the new formulation improves noticeably the fairness approaching the maximum, 1. This behavior can be easily explained with an example. Let focus on 6 users, in this case the original formulation allocates always 128 kbps to all users. Since the fair bit rate is 141.67 kbps, the fairness results in 0.90. On the other hand, the UB-HNN-DRA typically allocates 256 kbps to one user, 128 kbps to 4 users and 64 kbps to the remaining user. This allocation uses more bandwidth but is more unfair, with a fairness of 0.71. Nevertheless, the users with 256 and 64 kbps are selected randomly and, therefore, these users switch in different runs. In the long-term, i.e. when considering several resource allocation processes, users tend to be served with 138.67 kbps which is nearer to the fair bit rate. Note that Fig. 5.b) is Fig. 4 normalized by the maximum capacity. This fact can be easily concluded from (6) taking into account that the average total allocated bandwidth is equally divided between users in the long-term.

VII. CONCLUSIONS

This paper has underlined main drawbacks identified in an existing user-centric HNN-DRA formulation for packet-switched communication systems and has presented an enhanced formulation. The correct differential equation for evolution to the energy function minima has been derived and an efficient energy formulation, together with optimum weighting coefficients, presented. The paper has demonstrated that the proposed UB-HNN-DRA algorithm maintains the best bandwidth maximization and long-term fairness that can be reached with the original HNN-DRA and improves the behavior of the neural network reducing the oscillation probability to negligible values. These are critical aspects for the effective real-time provision of multimedia services over future wireless systems to a large number of users with varying QoS requirements.

References

[1] L. Xu, X. Shen and J.W. Mark, Dynamic Fair Scheduling With QoS Constraints in Multimedia Wideband CDMA Cellular Networks, in: IEEE Trans. Wireless Communications, Vol. 3, No. 1, January 2004.

[2] S. J. Oh, D. Zhang and K.M. Wasserman, Optimal Resource Allocation in Multiservice CDMA Networks, in: IEEE Trans. Wireless Communications, Vol. 2, No. 4, July 2003.

[3] M. Soleimanipour, W. Zhuang and G.H. Freeman, Optimal Resource Management in Wireless Multimedia Wideband CDMA Systems, in: IEEE Trans. Mobile Computing, Vol. 1, No. 2, April-June 2002.

[4] D. Zhao, X. Shen and J.W. Mark, Radio Resource Management for Cellular CDMA Systems Supporting Heterogeneous Services, in: IEEE Trans. on Mobile Computing, Vol. 2, No. 2, April-June 2003.

[5] C.W. Ahn, and R.S. Ramakrishna, QoS Provisioning Dynamic Connection-Admission Control for Multimedia Wireless Networks Using a Hopfield Neural Network, in: IEEE Trans. on Vehicular Technology, Vol. 53, No. 1, Jan. 2004.

[6] N. García, R. Agustí and J. Pérez-Romero, A User-Centric Approach for Dynamic Resource Allocation in CDMA systems based on Hopfield Neural Networks, in: Proc. IST Mobile and Wireless Communications Summit, Dresden, Germany, 2005.

[7] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, in: Proc. Nat. Acad. Sci., Vol. 81, May 1984.

[8] J.J. Hopfield and D.W. Tank, Neural computation of decisions in optimization problems, in: Biological Cybernetics, Vol. 52, 1985.

[9] S. Abe, Global Convergence and Suppression of Spurious States of the Hopfield Neural Networks, in: IEEE Trans. Circuits and Systems, Vol. 40, No. 4, April 1993.

Figure 1. Simulation example of the neurons evolution with an infinite shape parameter.

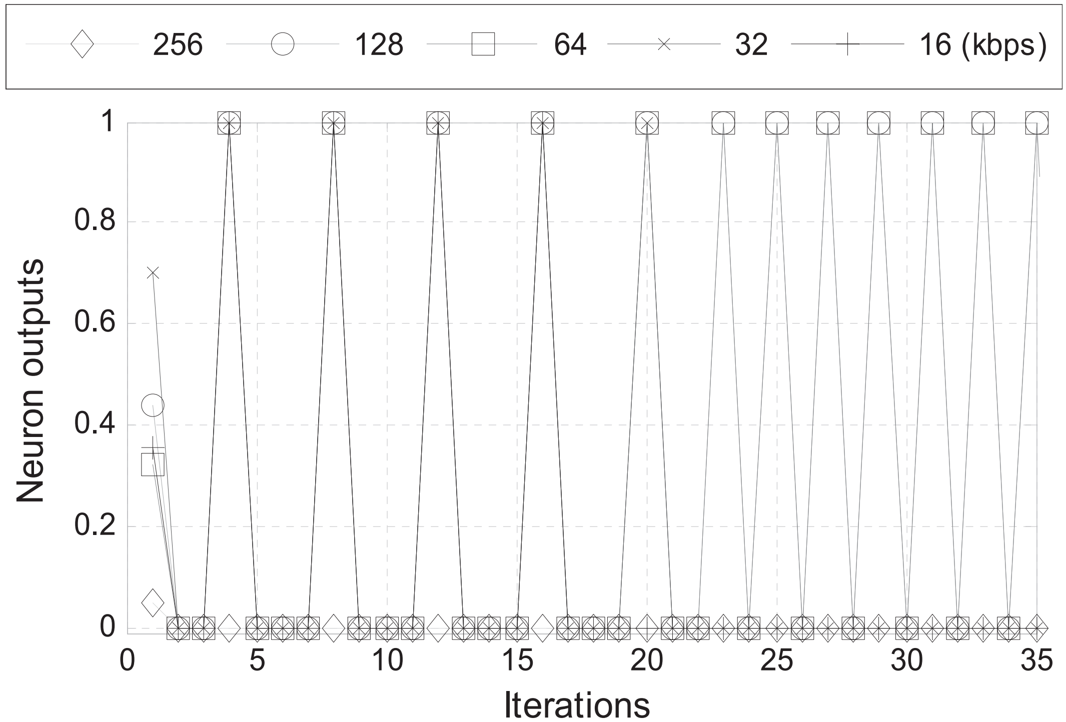

Figure 2. Simulation example of the two energy functions studied in this paper.

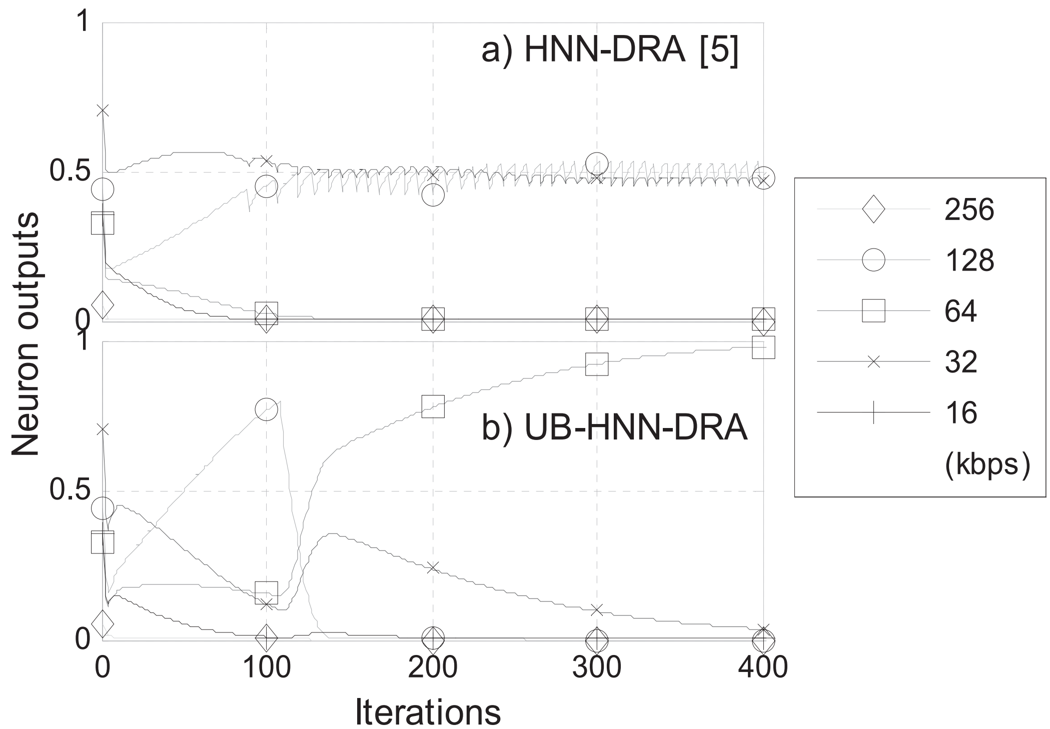

Figure 3. HNN oscillation probability as a function of the system load.

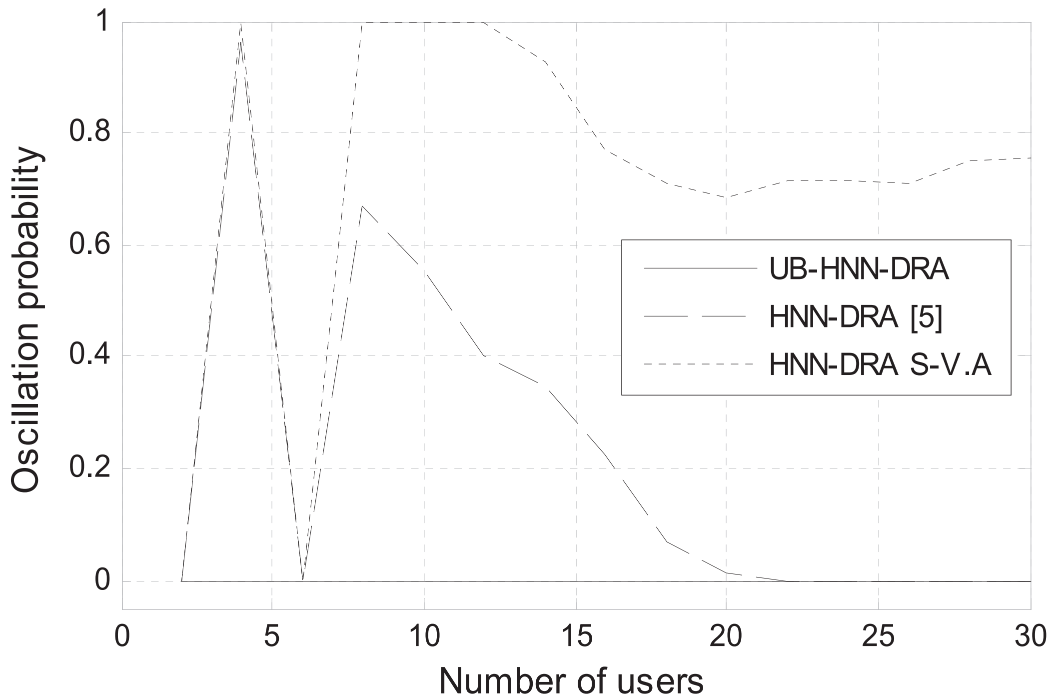

Figure 4. Average total allocated bandwidth as a function of the system load.

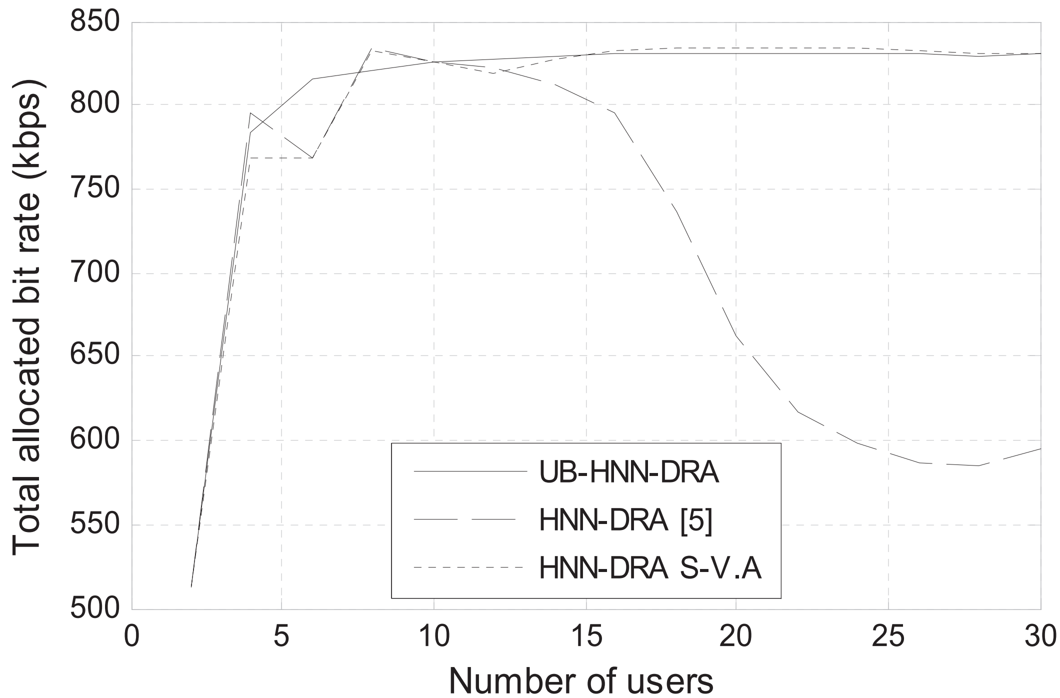

Figure 5. Fairness. a) Average instant fairness. b) Long-term fairness.