

# Neural Self-Organization for the Packet Scheduling in Wireless Networks

Leonardo Badia, Michele Boaretto, Michele Zorzi  
Dipartimento di Ingegneria, Università di Ferrara  
via Saragat, 1 – 44 100 Ferrara, Italy  
{lbadia, mboaretto, zorzi}@ing.unife.it

**Abstract**— This paper aims at introducing possibilities of self organization for packet-switched networks at the scheduling management level. By discussing already existing scheduling strategies, we identify several contrasting needs that should be jointly addressed. The employment of the same algorithm for the whole network leads to low performance. On the other hand, adjusting the management cell-by-cell is not feasible and requires coordination. Hence, we propose a general framework for the scheduler, which can be easily tuned by means of a neural network. In this way, cells are grouped and self-similarities are identified, so that the differentiation in the management is lighter and a significant performance improvement can be achieved. Moreover, a general tunability of the system is introduced, allowing to cut the right trade-off between system complexity and QoS in a simple way.

## I. INTRODUCTION

Wireless communication systems are expected to offer a wide range of rich multimedia services. This implies high and time-varying requirements in terms of bandwidth, and heterogeneous behavior in terms of QoS constraints (mainly expressed by BER and delay tolerance). The traffic load is usually considered to be concentrated essentially in the downlink (where the capacity bottleneck is therefore expected). Finally, a packet-switched communication is generally assumed.

In this context, efficient radio resource allocation needs advanced algorithms [1] for the packet scheduling. Some multimedia applications present certain QoS elasticity: the constraints might be sometimes relaxed (soft QoS concept) and the extent and duration of this relaxation can be directly related to the service and, moreover, to the user charge plan. The scheduler must consider the system flexibility [2], in order to enhance the packet allocation. For example, Non Real Time (NRT) services allow a small degradation even for all users in order to admit the scheduling of another packet source, which results in a more efficient resource usage.

In general, the scheduling must be performed by an advanced Radio Resource Manager block. In wireless cellular systems, it is possible to identify a trade-off between two different strategic choices: to prioritize the transmission of users with good channel state, or to encourage fairness among the users. The former leads to the well-known channel-state-dependent scheduling (indicated in the following as “C/I”) [3], whereas the latter can be achieved even with simple strategies but in general associated with low peak performance [4]. This means

that an efficient allocation is not trivial. From the operator’s point-of-view, this implies a choice between two possibilities, both unpleasing: to have different QoS levels for users possibly having the same tariff and the same quality constraints, or to miss some opportunity to allocate high data rates.

A joint strategy [5] to consider both issues and the tunability between fairness and channel state should be considered. This can be done by considering the perspective of the service provider, that prefers to maximize the number of satisfied users. This objective does not always agree with the maximization of the system efficiency.

Moreover, one should consider that in general, users’ service appreciation depends on different aspects of the resource management. To adopt the same strategy for the whole network might lead to an inefficient resource usage. On the other hand, exchange of information to regulate the scheduling in a cell-wise manner implies a large and probably unnecessary overhead. Besides this, also the mobility would severely affect the performance, as users experience an unjustifiably high difference of treatment before and after every handover.

To manage these problems, a novel strategy is presented, called Neural Self-Organizing Map (NSOM) algorithm, in which the trade-off between throughput and fairness is cut by means of an auto-tunable neural strategy [6]. The cells are clustered by considering representative parameters which allow the identification of similarities. In this way the operator is able to setup the network in a simple manner, without excessively increasing the computational complexity. In particular, in this work we apply the NSOM algorithm to a UMTS High Speed Downlink Packet Access (HSDPA) channel, by showing performance improvements which are in particular significant for achieving high users satisfaction, which is what the system is designed for.

This paper is organized as follows: in Section II the trade-off between fairness and QoS is stated. In Section III the NSOM algorithm is proposed as a possible solution. Its main features are described and its application to the case study is discussed. In Section IV simulation results are presented to validate the algorithm. Finally, Section V concludes the paper.

## II. THE GENERAL TRADE-OFF BETWEEN PEAK PERFORMANCE AND FAIRNESS

Considering the instantaneous wireless channel conditions is a key task, as location-dependent and bursty errors are typi-

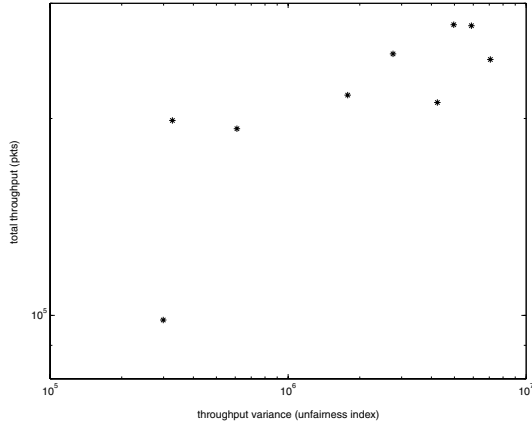


Fig. 1. throughput-fairness dispersion graph - C/I heuristic

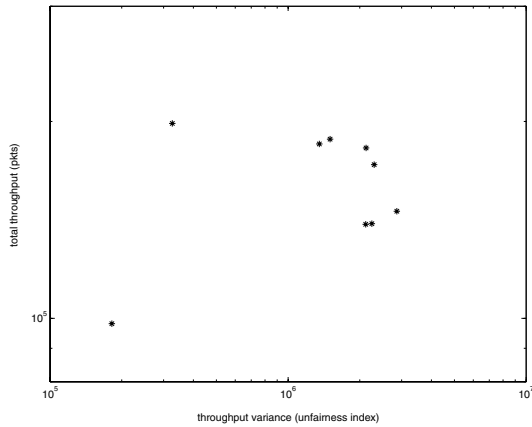


Fig. 2. throughput-fairness dispersion graph - LSF heuristic

cal. A user in a fading dip may experience a bad radio condition and may be unable to transmit. The scheduling framework has to take into account the channel conditions and to give priority to users which experience a good channel.

However, a joint approach to fairness and C/I is necessary; hence, it is important to identify how to tune the performance between these two contrasting objectives. In Figures 1 and 2 we present a possible way to analyze the trade-off. Here, simple scheduling strategies have been considered to depict the extreme cases: a pure C/I strategy, and a strategy where the least served user is labelled with the maximum priority, called Least served first (LSF). This latter strategy aims at emphasizing fairness of the system. We used the dispersion graphs to represent the behavior of these algorithms, by selecting the variance of the throughput as a measure of unfairness. On the vertical axis instead, the total throughput is plotted to depict the efficiency of the scheduler. Different points on the graphs correspond to simulations with a different number of users. The way in which the points are dispersed in the plane might be considered descriptive of the strategy behavior. It is clear from the Figures that with the existing algorithms one can only obtain good throughput but low fairness or vice versa. However, how to adjust them in a tunable way is still an open issue.

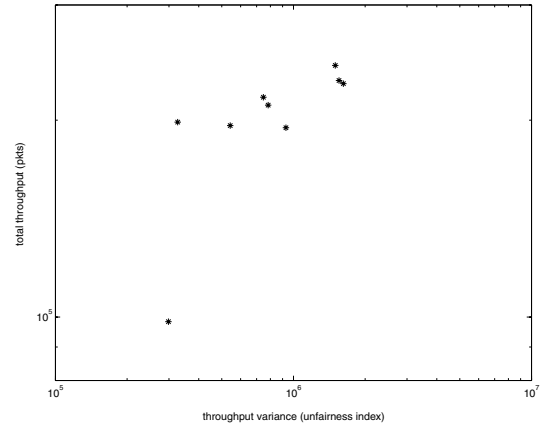


Fig. 3. throughput-fairness dispersion graph - LC heuristic

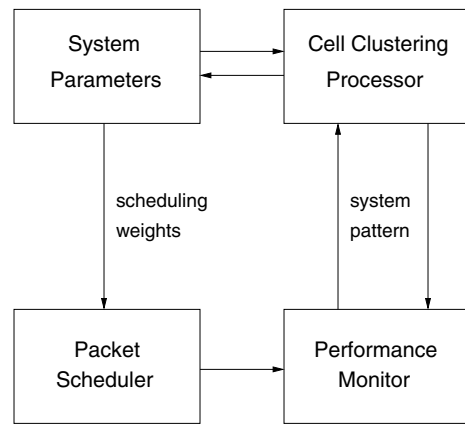


Fig. 4. System Architecture for Tunable Management

A possibility to obtain a *Tunable Scheduling Strategy aimed at QoS* is achieved by considering a Linear Combination Algorithm, as done in [7], which enables us to take into account contrasting requirements. As can be seen in Figure 3, this strategy allows to improve the performance. In particular, the dispersion points are closer to the top-left part of the graph, which is the more suitable for the users, as it means high throughput but also a satisfactory degree of fairness. However, the propagation condition changes in a quick manner, hence the parameters of the linear combination may no longer be valid after a certain amount of time. To follow the variations and to account for a reactive system, we consider the framework of Figure 4, where the general scheduling strategy discussed above is presented, and the scheduler weights are dynamically adjusted by means of a QoS-driven network monitoring process [8], as shown in Figure 4. This procedure is assumed to be implemented in the same manner in each cell. However, the scheduling weights and the feedback functions might or might not be the same. In general, a trade-off must be cut between adopting the same procedure for the whole network or a different strategies for each cell.

### III. THE NSOM ALGORITHM

The basic idea of Neural Self-Organizing Map (NSOM) is simple yet effective. The trade-off between applying the same

scheduler to the whole network or specializing scheduling for each cell can be cut by means of clustering. If the cells are clustered in a manner representative of their similarities, the overhead increasing can be reduced. This is especially true when the network size becomes considerably high.

To obtain the clustering, the NSOM algorithm defines a mapping from high dimensional input data space onto a regular two-dimensional array of neurons. Every neuron  $i$  of the map is associated with an  $n$ -dimensional reference vector, where  $n$  denotes the dimension of the input vectors.

The reference vectors together form a codebook. The neurons of the map are connected to adjacent neurons by a neighborhood relation, which dictates the topology, or the structure, of the map. The most common topologies in use are rectangular and hexagonal. Adjacent neurons belong to the neighborhood  $N_i$  of the neuron  $i$ .

In the basic NSOM algorithm, the topology and the number of neurons remain fixed from the beginning. The number of neurons determines the granularity of the mapping, which has an effect on the accuracy and generalization of the NSOM. On the other hand, note that this is a degree of freedom for the provider to tune the computational complexity of the clustering. During the training phase, the NSOM forms an elastic net that folds onto the “cloud” formed by input data. The algorithm controls the net so that it strives to approximate the density of the data. The reference vectors in the codebook drift to the areas where the density of the input data is high.

Eventually, only few codebook vectors lie in areas where the input data is sparse. The learning process of the NSOM goes as follows:

- 1) One sample vector  $\mathbf{x}$  is randomly drawn from the input data set and its similarity (distance) to the codebook vectors is computed by using e.g. the common Euclidean distance measure.
- 2) After the Best Matching Unit (BMU) has been found, the codebook vectors are updated. The BMU itself as well as its topological neighbors (i.e. neurons in a defined radius distance from the BMU) are moved closer to the input vector in the input space, i.e., the input vector attracts them. The magnitude of the attraction is governed by the learning rate ( $\alpha$  parameter). As the learning proceeds and new input vectors are given to the map, the learning rate gradually decreases to zero according to the specified learning rate function type. Along with the learning rate, the neighborhood radius decreases as well.

The update rule for the reference vector of unit  $i$  ( $M_i$  is the  $i$ -th neuron weight) is the following:

$$M_i(t+1) = M_i(t) + h(t)d(x(t), M_i(t)) \quad (1)$$

where  $x(t)$  is the input data vector,  $d(\cdot)$  is the Euclidean distance between the two vectors, and finally  $h(t)$  is the bubble neighborhood function. This function updates neighborhood neurons at each training input step (training data  $x(t)$ , neuron  $n_i$ , repeated for each neighborhood neuron  $n_j$ ), as:

$$h(t) = \alpha(t)e^{-(d(x(t), M_i(t))^2)/(2d(n_i, n_j)^2)}; \quad (2)$$

The parameter  $\alpha$ , called *learning rate*, decreases linearly at

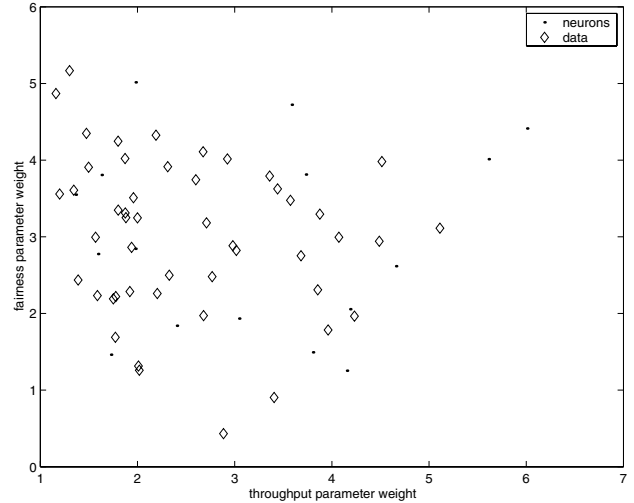


Fig. 5. Neuron vs. data

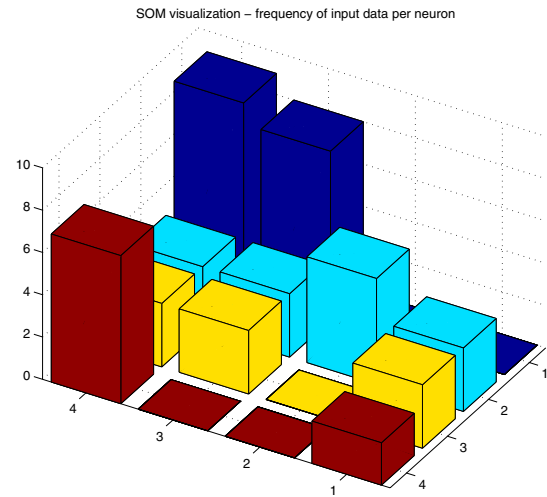


Fig. 6. Data neuron frequency

each step:

$$\alpha = \alpha_0(1.0 - (t/T)) \quad (3)$$

The neighborhood of a neuron is determined by means of a parameter  $\rho$ , which is also linearly decreased:

$$\rho = \rho_0(1.0 - (t/T)) \quad (4)$$

so that  $n_j$  is in the neighborhood of  $n_i$  if  $d(n_i, n_j) < \rho$ .

Steps 1 and 2 together constitute a single training step and they are repeated until the training ends. The number of training steps  $T$  must be fixed prior to training the NSOM because the rate of convergence in the neighborhood function and the learning rate is calculated accordingly. After the training is over, the map should be topologically ordered. This means that  $n$  topologically close (using NSOM distance measure, e.g., Euclidean) input data vectors map to  $n$  adjacent map neurons or even to the same single neuron.

The coefficients of the LC algorithm are hence tuned neuron by neuron, by considering a uniform way to cut the tradeoff between throughput and fairness. The linear combination of the scheduler algorithm is adjusted accordingly, in order to reduce

TABLE I  
ACTIONS FOR THE AUTOTUNING POLICY

Throughput	Fairness	Action
++	++	ok
++	**	ok
++	--	>
**	++	<
**	**	ok
**	--	>
--	++	<
--	**	<
--	--	<

the variations. An example of the autotuning policy may be the one sketched in Table I. Here, if differences higher by a fixed amount (which in the case study has been considered equal to the 15%) are detected between a neuron and the average value of its neighborhood, the coefficients can be changed. The decision are taken depending on the values being above (++), below (--) or within (\*\*) the admitted range centered in the average value. Adjustments are made with the finest granularity, sufficient to have convergence to a uniform solution (in the case study we considered a 5% increase). The "<" action means an increase of the C/I coefficient with respect to the Least-Served-First coefficient, whereas the ">" means the opposite and "ok" means no changes. The algorithm of Table I slightly prioritizes throughput over fairness but other choices are still possible. For more details about possible choices, see [11].

Samples of results for the NSOM algorithms are represented in Figures 5 and 6. Figure 5 shows the final configuration of the neuron approaching the data plotted on the same graph. Figure 6 instead, shows the frequency of input data given to each neuron.

In this paper we apply this strategy to a UMTS - High Speed Downlink Packet Access (HSDPA) network. In these network, the traditional scheduling methods are similar to C/I scheduler, i.e., the channel state is used as decision parameter to sort users to schedule. In more advanced schemes others information such as queue state and head packet delay are used to obtain a trade off between throughput and fairness performance [7]. All the information are combined together in a weighted manner to obtain a scheduling priority parameter. There are several possibilities to define such a combination. To keep the framework as general as possible, we will speak in the following of a simple combination in which there are two components (possibly containing more terms), which can be related to the C/I effect improvement or to the LSF component, so that the point of trade-off between Throughput and Fairness can be adjusted by tuning the parameters accordingly. In particular, increasing the C/I weight has, theoretically speaking, the same effect than decreasing the LSF one. In traditional systems scheduling weights are yet fixed a priori; hence, the decision on the relative weights is taken at the beginning and kept even if it is inefficient. Rather, in this paper we apply the NSOM strategy to implement an autotuning system in which, after a data monitoring task, weights are updated to be adapted to the current state of the network,

TABLE II  
PARAMETERS OF SIMULATION SCENARIO

Parameter (symbol)	value
cell radius ( $d$ )	250 m
number of cell ( $K$ )	49
Hata path loss exponent ( $\alpha$ )	3.5
shadowing parameter ( $\sigma$ )	4dB
Doppler frequency ( $f_d$ )	2Hz
gain at 1 m ( $A$ )	-30dB
SIR target ( $\gamma_t$ )	2.5dB
orthogonality factor ( $f$ )	0.3

as in Figure 4. In the first step network performance data as average throughput and fairness (measured by means of standard deviation of users' throughput) in each cell are collected and encoded to be examined by the autotuning subsystem. This system analyzes data and create a NSOM to find similarities between cell data and to obtain a clustering of the cells. After this step clusters are used to update cell scheduling system in an easy and faster way, due to a grouped regulation based on parameter similarities.

This might be useful in hot-spot scenarios in which we have a high correlation between a cell and its neighbors. Here the service provider has to efficiently allocate the radio resource, for different main reasons: the traffic is expected to be intense, and also the service is usually assumed to be highly demanding in terms of QoS, like video streaming or entertainment applications. Thus, the application of NSOM algorithm to such a situation can lead to an appealing simplification of the management for complex and congested networks.

#### IV. RESULTS

We use a UMTS - High Speed Downlink Packet Access (HSDPA) simulator, in which NSOM functionalities are applied to packet scheduling. The simulation parameters are reported in Table II. Note that the cells of the simulated area are wrapped around so that border effects are avoided.

The simulator includes performance monitoring, realized with a sliding window mechanism updated on a time-scale of the order of milliseconds [9]. The C/I algorithm is implemented, as well as the NSOM algorithm derived starting from a C/I algorithm applied to a neural clustered network, so that each cluster refreshes its scheduler weighting coefficients independently.

In Figure 7 the total throughput obtained by the network is compared, and it is shown that the NSOM algorithm is able to improve the performance up to 7% with respect to the C/I algorithm. Figure 8 shows the effects of application of NSOM strategy in terms of trade-off between throughput and fairness. It is emphasized that also the average throughput is increased by 6% with respect to the C/I algorithm, whereas the unfairness, measured by means of the standard deviation of the throughput is also increased, even though by a smaller factor (about 4%). This is of course justified since the clustering allows a more suitable scheduler for each cell. Thus, in this case the increase of the unfairness index should not be seen only as a negative ef-

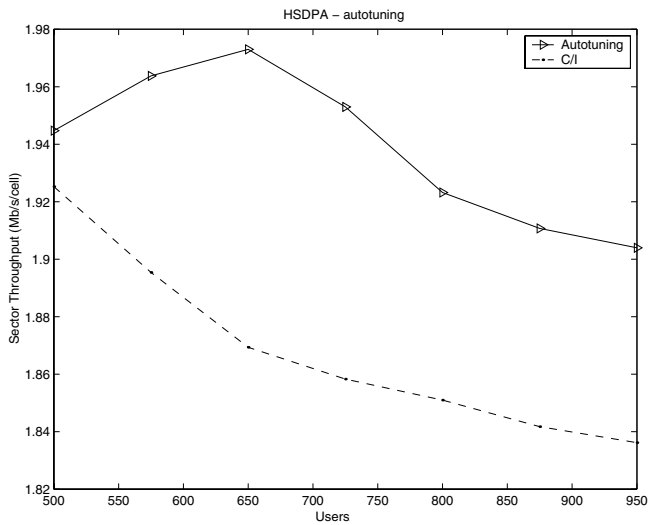


Fig. 7. Total throughput

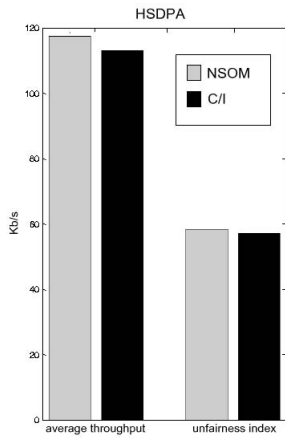


Fig. 8. Average throughput vs. throughput st.dev

fect, as it implies also that the network management is adapted in a more close way to the differences present within the network.

Finally, Figure 9 shows the improvement achieved by NSOM algorithm in terms of users' satisfaction. Here, the operating point for two metrics is represented, the former being the users' satisfaction rate, whereas the latter is the revenue obtained with a flat pricing strategy. These two metrics are related, as the satisfaction is measured as the number of users which meet the SIR target, divided by the total number of users. The revenue is obtained by considering as paying users only those which not only meet the SIR target, but also a higher threshold, 0.5dB higher. In this case, the same tariff is paid by the users. Else, no fee is required as the offered QoS is considered too low.

As a matter of fact, the revenue in this Figure is proportional to the number of users which perceive even a higher satisfaction degree. It might be seen that the NSOM procedure is able to improve the management so that the number of satisfied users is improved by 6%, but this improvement is more significant

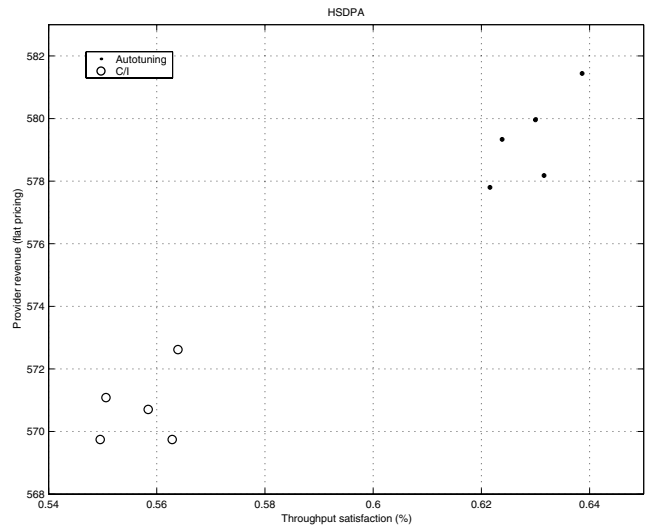


Fig. 9. Revenue vs. Satisfaction rate

by considering only the paying users (which is a higher SIR requirement), as the revenue is increased up to 16%. This ultimately justifies the use of such a strategy for an operator, due to its possibility of guaranteeing higher data rates especially for services which require higher QoS.

## V. CONCLUSIONS

Algorithms for packet scheduling can be improved with different strategies. The added overhead and complexity must be justified by the gain obtained by introducing new features to the scheduler. However, for multimedia network it is important that the provider, besides improving the performance, is also able to tune the behavior of the network. Moreover, due to economic considerations, which we addressed also from the technical side, the users' satisfaction is another important aspect.

In this paper we achieve improvements by means of the Neural Self-Organizing Map algorithm, which offers a general framework that can be easily adjusted to different topologies, as it makes the network self-tunable. The results show that the application of a classical efficient strategy, like the C/I scheduler, can be suitable for small networks where there is no need for coordination or differentiation. On the other hand, when the network size increases, our strategy is able to achieve a performance gain by grouping cells into clusters. Beyond the performance gain in terms of throughput, which heavily depends on network size and number of users, there is also a large increase in terms of percentage of satisfied users.

A tunable scheduling algorithm could be a good and portable solution even for networks characterized by the presence of hot spots, rather than locally centralized solutions, which require a heavy interchange of information with the rest of the network. Moreover, the virtual clustering mechanism is an efficient solution to counteract the increased computational complexity of the system. Thus, our scheme can be of significant interest for large-scale networks with challenging QoS requirements.

As a general conclusion, the QoS of the network is improved and several further observations open up about the improvement gained for the network welfare (which ultimately affects the operator's revenue). Finally, the operator gains another degree of freedom in the service supply, as the proposed algorithm offers a way to control, in a simple but effective way, several parameters like system complexity and QoS offered to the users.

#### REFERENCES

- [1] M.A. Arad, A. Leon-Garcia, "A generalized processor sharing approach to time scheduling in hybrid CDMA/TDMA", *Proceedings of INFOCOM 98*, 17th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, pp 1164–1171, 1998.
- [2] C. Comaniciu, N. Mandayam, D. Famolari, P. Agrawal, "Admission and flow control for multimedia CDMA", *Proceedings of ICME 2000*, IEEE International Conference on Multimedia and Expo, 2000. Volume 3, pp 1265–1268, 2000.
- [3] P. Bhagwat, P. Bhattacharya, A. Krishna, S.K. Tripathi, "Enhancing throughput over wireless LANs using Channel State Dependent Scheduling", *Proceedings of INFOCOM '96*, Volume 3, pp 1133–1140, 1996.
- [4] V. Bharghavan, S. Lu, T. Nandagopal, "Fair Queuing in Wireless Packet Networks: Issues and Approaches", *IEEE Wireless Communications*, Volume 6, Issue 1, pp 44–53, 1999.
- [5] A. Stamoulis, G.B. Giannakis, "Packet Fair Queuing scheduling based on multirate-transparent CDMA Wireless Networks", *Proceedings of INFOCOM '00*, Volume 3, pp 1067–1076, 2000.
- [6] K. Raivio, O. Simula, J. Laiho, "Neural Analysis of Radio Access Network," *IEEE ICDM*, 2001.
- [7] D. Angelini, M. Zorzi, "Channel adaptive fair scheduling for CDMA/TDD downlink packet communications," *Proceedings of PIMRC 2002*, volume 5, pp 2419–2423, 2002.
- [8] C. Lindemann, M. Lohmann, A. Thümmler, "Adaptive Performance Management for UMTS Networks", *Computer Networks*, Volume 38, pp 477–496, 2002.
- [9] H. Zhang, "Service disciplines for guaranteed performance service in packet-switched networks," *Proceedings of the IEEE*, Volume 83, pp 1374–1396, 1995.
- [10] A. Ultsch, H.P. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," *Proceedings of the International Neural Network Conference*, pp 305–308, Dordrecht, the Netherlands, 1990.
- [11] J. Laiho, A. Höglund, T. Buot, "RAN Autotuning and Advanced Monitoring Methods," in: "Radio Network Planning and Optimisation for UMTS," by J. Laiho, A. Wacker, T. Novosad (eds.), John Wiley and Sons, Baffins Lane, UK, 2002.