A Management Scheme for Distributed Cross-Layer Reconfigurations in the Context of Cognitive B3G Infrastructures

G.Dimitrakopoulos[1], K.Tsagkaris[1], K.Demestichas[2], E.Adamopoulou[2], P.Demestichas[1]

[1]University of Piraeus, Department of Digital Systems, 80 Karaoli Dimitriou str., 18534, Piraeus, GREECE, Tel: +30 210 414 2760, Fax: +30 210 414 2744, e-mail: gdimitra@unipi.gr

[2]National Technical University of Athens, School of Electrical and Computer Engineering, Athens, GREECE

Abstract. Current research efforts in wireless communications are targeted at the evolution of B3G (Beyond the 3rd Generation) wireless infrastructures. The operation of B3G infrastructures envisions dynamic adaptations to external stimuli, which can be facilitated through the exploitation of cognitive networking potentials. Cognitive networks dispose mechanisms for dynamically selecting their configuration (algorithms and parameter values, at different layers of the protocol stack), through appropriate management functionality that takes into account the context of operation (environment characteristics and requirements), profiles, goals, policies and knowledge that derives from previous experience. This paper focuses on such management functionality and it addresses a problem, dealing with "*Distributed, Cross-Layer Reconfigurations" (DCLR)*, which aims at assessing and selecting the most appropriate configuration per network element in a cognitive network. In essence, this work contributes in four main areas. First, a fully distributed formulation and solution to the DCLR problem is provided, which is important for the management of a particular reconfigurable element in a cognitive context. Second, robust learning and adaptation, strategies are proposed, for estimating and gaining knowledge of the performance potentials of alternate reconfigurations. Third, a computationally efficient solution to the problem of exploiting the performance potentials of reconfigurations is provided, in order to rate reconfigurations and finally select the best ones. Finally, results that expose the behaviour and efficiency of the proposed schemes, are presented.

Keywords. Cognitive networks, B3G wireless infrastructures, Cross-layer optimization, Utility, Learning and Adaptation, Autonomic computing, Bayesian networks

1. INTRODUCTION

Immense research and development effort is dedicated to the development of new wireless networking technologies. This work delivers powerful and affordable, *high-speed*, *wireless access* solutions. Currently, the wireless landscape includes: (i) mobile communications (2G/2.5G/3G/3.5G); (ii) wireless local/metropolitan area networks (WLANs/ WMANs); (iii) wireless personal area networks (WPANs) and short range communications, (iv) digital video/audio broadcasting (DVB/DAB). This infrastructure derives from work conducted in the context of research programs, forums and standardisation bodies [1,2,3,4,5,6].

Moreover, the wireless world is migrating towards the era of beyond the 3rd generation (B3G) [4] communications. The main idea of the B3G vision is that a network operator (NO) can rely on different Radio Access Technologies (RATs), for achieving the required quality of service (QoS) levels, cost-effectively, thus increasing their exploitation. Figure 1 shows a B3G infrastructure that includes numerous, versatile RATs. In principle, major part of the infrastructure will be owned by a NO, and some components will be owned by affiliated NOs (e.g., WLANs or the broadcasting component). Each RAT has certain capabilities, in terms of achievable bit rate, coverage, mobility support and cost. A NO can select those that are best suited for delivering the desired bit rates and associated QoS levels, given the current environment conditions and requirements. For users, the supporting network technologies are irrelevant, as long as cost and business criteria (QoS, reliability, etc.) are met.

A manner to facilitate this cooperation amongst networks [7,8] is to design B3G systems capable of reconfiguring their infrastructure and operational parameters [9,10], so as to respond to the changing contextual conditions. Reconfigurability, that also allows for the over-the-air download and installation of the Sofware Defined Radio (SDR) oriented modules that are necessary for the operation of the selected RAT [11], promises a significant reduction in the infrastructures' Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) [12].

However, the accumulation of such novel features in legacy and emerging technologies makes B3G environments really complex. An interesting option proposed for handling complexity, is the integration of the "learning" factor in the behaviour of B3G infrastructures, which can endow them with "cognitive networking" capabilities [13,14,15,16,17,18]. In general, cognitive systems are able to obtain knowledge from previous interactions with the environment, and base their future behaviour on this knowledge, other goals and policies, so as to adapt to external stimuli and optimize their performance. Cognition may lie in the radio part of a system (cognitive radio) [15,19] or be distributed over the radio, O&M and switching subsystem of a network [13,17].

Accordingly, cognitive networks benefit from the existence of reconfigurable platforms which enable dynamic changes in their configuration, while they also require intelligent *management functionality*, in charge of selecting the most appropriate configurations. This paper provides such functionality, by addressing an important problem for the management of a reconfigurable element, which operates and is managed in parallel with other elements. The problem is called "*Distributed, Cross-Layer Reconfigurations" (DCLR)* and it results in a cross-layer optimization of an element, as will be shown. This work confronts two main challenges.

- The provision of stable and reliable learning and adaptation strategies for discovering the potential capabilities of candidate reconfigurations, in a distributed manner. The proposed method complements legacy discovery functionality [20] and relies on Bayesian networks [21].
- The provision of a computationally efficient solution to the problem of exploiting the capabilities of candidate reconfigurations. This exploitation yields a rating of the candidate reconfigurations, and leads to the selection of the best one i.e. the one that results in the provision of the highest possible QoS levels.

Our focus is on a *highly distributed* problem version, thus resulting in a scheme that requires minimal interactions with other elements. Eventually, our goal is to employ autonomic computing concepts [13,22,23,24]. This work is complementary to areas like software radio [15,25] and spectrum management [15,26], which essentially provide results that enhance our portfolio of potential reconfiguration decisions. The organization of the rest of the paper is as follows. Section 2 presents the motivation for this work, including a high level description of the DCLR problem. Section 3 formulates the DCLR problem, and outlines the two main parts of its solution. Sections 4 and 5 describe the two parts of the problem's solution. Section 6 provides results that showcase the behaviour of the proposed scheme, and section 7 includes concluding remarks.

2. Motivation and High Level Description of the DCLR Problem

This section describes the basic features of cognitive networks, i.e. (a) the reconfigurable platforms and (b) their management functionality. Additionally, it provides a high level description of the problem that such functionality aims at solving, which also constitutes the main focus of this paper.

2.1. Reconfigurable Platforms

Figure 2 (a) presents an example on the role of reconfigurable platforms [27] in cognitive networks. We consider a network segment that complements the B3G infrastructure (e.g., the one depicted in Figure 1). Traditional approaches either load a set of pre-defined configurations, or use a policy server to dole out policies as appropriate. This paper shows how these approaches can be improved by enabling each element of the segment to *dynamically* select its own (re)configuration. Sample reconfigurations include (at the physical and medium access control layers) the selection upon the RAT and spectrum for operation. So, each element may:

(i) change the RAT it operates with and maintain the spectrum

(ii) maintain the RAT and change the spectrum

(iii) change both, RAT and spectrum

(iv) maintain both, RAT and spectrum (no reconfiguration is implemented)

In Figure 2(a), three RATs and five frequencies are involved in the configuration of elements e_1 and e_2. The elements use RATs R1 – R2 and frequencies F1 – F4 for handling the environment conditions in time zone n. Time zone $n+1$ is an example of all the reconfigurations that may occur. In element e_1, the configuration of one transceiver remains intact (R1, F1), while the other transceiver maintains the RAT (R2) and changes the frequency (replaces frequency F2 with F3). In element e_2, one transceiver changes only the RAT (from R2 to R1) and maintains the frequency (F4), while the other changes its configuration completely from (R1, F3) to (R3, F5).

In the light of the above, each element may be multi-standard. A subset of technologies is used, namely those that are most appropriate for the context of operation. Network layer reconfigurations accompany the changes at the lower layers. At this layer each configuration includes the algorithms and parameters for routing and congestion control, and in general, the pattern for interconnection with other elements of the network. Additionally, reconfigurations can be extended to the applications layer, through specifying the QoS levels of the applications. In general, reconfigurations are software defined. Specifically, each element is controlled by management functionality that has to solve a problem of cross-layer flavour. This is the subject of the next subsection.

2.2. Management Functionality: the DCLR problem

In general, the operation of the management functionality for cognitive networks can be depicted as in Figure 2(b) [17]. Specifically, the network continuously observes the environment, looking for potential changes that can affect its operation. Observations form the basis for initiating machine-based analysis (reasoning) to see if the reconfiguration process should be invoked. Once the decision is taken, the network acts accordingly. This loop can be augmented by a machine learning process [28], which leads to cognition. The loop is guided by a set of policies and goals.

Figure 3 provides the overall description of the management functionality (DCLR) proposed in this paper. Environment observations are done through the monitoring and discovery procedures. Monitoring is targeted to the service area requirements and to the performance of the current configuration of the element. Discovery is targeted to the capabilities of alternate configurations. Machine learning techniques are used for enhancing the robustness of the discovery procedure, which can be the most vulnerable, because of the stochastic nature and time variance of the radio environment. The integration of such techniques is an essential first step that justifies the cognitive nature of our schemes. Finally, our management functionality takes into account profiles and policies regarding both the managed elements and served users.

The proposed management mechanisms undertake decisions that affect the protocol stack in a cross-layer fashion. The next section, accordingly, formulates mathematically the DCLR problem and describes in detail the input and the output (decisions for element reconfiguration); sections 4 and 5 present the solution.

3. DCLR PROBLEM FORMULATION

3.1. DCLR input

The input of the DCLR problem is classified in three main categories: (i) monitoring, (ii) discovery, and (iii) profiles.

Monitoring. This part gathers monitoring information for estimating traffic requirements and mobility characteristics in the service area. Set U_e denotes the set of users in the service area of e. The set of services (applications) requested is S_e. It is assumed that each user $u \in U_e$ requests a service, $s(u) \in S_e$. This model covers physical users that require more than one service. Regarding mobility, assuming a semi-stationary state has been reached, we can associate with each user u a location $l(u)$. This is the subset of the service area, of reconfigurable element e, in which u is found.

Discovery. This part exploits basic monitoring information for estimating the capabilities (achievable bit rate and coverage) of the candidate configurations. These capabilities can change over time, as they are influenced by the changing conditions in the environment, especially the behavior of "near-by" elements. All elements act in a completely distributed (autonomic) manner. This poses a significant engineering challenge: how to increase the degree of assurance that, by assigning a certain configuration, $c=(r, f)$, to transceiver t of e, the resulting maximum achievable bit rate, $abr_e(c)$, and coverage, $cov_e(c)$, will be, e.g., x Mbps and y km, respectively. A probabilistic model, as well as a learning and adaptation strategy, should be adopted. The resulting problem is: *Given a specific candidate configuration $c=(r, f)$, how can we predict the most probable values of the random variables achievable bit rate and coverage?*

Section 4 solves this problem through a robust, learning and adaptation, strategy, based on Bayesian networks [21]. The solution does not violate the autonomy of network elements. In fact, no cooperation (e.g., no message exchange) between the different network elements is needed. Our schemes complement legacy schemes for the sensing (initial discovery) of the performance potentials of reconfigurations. In a similar manner, these learning and adaptation strategies can be applied in other parts of the problem input (e.g. the profiles).

However, the discovery part of the problem input is chosen as being more unstable (e.g. according to cognitive radio, entities might identify and jump to unused spectrum [15]).

Profiles. This part provides information on the candidate configurations of the element (element profiles). In general, an arbitrary reconfigurable element, e, will have a set of transceivers, T_e. Each transceiver, $t \in T_e$, will be capable of operating a set of RATs, $R_e(t)$. Moreover, there is a set of spectrum carriers, $F_e(t,r)$, with which t can operate RAT r. In general, for all $t \in T_e$, $F_e(t,r) \subseteq F_r$, where F_r is the set of spectrum carriers, with which RAT r can be operated, e.g., due to regulation or technological reasons. Each transceiver, t, has a set of candidate configurations, $C_e(t)$. Each configuration, $c \in C_e(t)$, is a set (r, f), where $r \in R_e(t)$ and $f \in F_e(t,r)$. Finally, the union of the sets above readily leads to the aggregate sets of RATs R_e, carriers F_e, and configurations C_e, which can be used in e.

Moreover, this part also describes the profiles, namely preferences, requirements, constraints of user classes, applications and terminals, as well as the policies and agreements of the NO. Set $Q(s,u)$ provides the target QoS levels at which service s ($s \in S_e$) should be offered to user u ($u \in U_e$). Set $R(s,u)$ specifies the set of RATs, through which service s can be offered to user u. The provision of service s, at QoS level q, to user u, is associated with a utility volume (importance) [29,30], $uv(s,q,u)$.

3.2. DCLR output

The reconfiguration decisions affect various protocol stack layers, fall into three main sets: (i) transceiver reconfiguration (physical / MAC layer); (ii) traffic distribution (network layer); (iii) QoS assignment (application layer).

Transceiver reconfigurations. These are denoted as A_{TC} = $\{c_e(t) \mid t \in T_e\}$. Each element of the set, $c_e(t)$, is the overall reconfiguration of transceiver t, and corresponds to a pair $[r_e(t), f_e(t)]$. Functions $r_e(t)$ and $f_e(t)$ are the RAT and spectrum, respectively, allocated to transceiver t. Each element of the set should respect the relation $c_e(t) \in C_e(t)$, or equivalently, $r_e(t) \in R_e(t)$ and $f_e(t) \in F_e[t, r_e(t)]$. This guarantees compliance with the element profiles (i.e., permissible allocations of RATs and spectrum to transceivers).

Traffic distribution. Set A_{TU} =$\{u_e(t) \mid t \in T_e\}$ expresses the new traffic distribution, due to the reconfiguration. Each element of A_{TU}, $u_e(t) \subseteq U_e$, includes the users that will be served by transceiver t of element e. For each $u \in u_e(t)$, the relation $r_e(t) \in R(s(u),u)$

should hold. This guarantees the provision of applications through permissible RATs, in accordance with the (terminal) profiles, NO policies and agreements. In addition, the relation Function $nr(\ldots)$ is kept flexible, in order to ensure applicability with various RATs. $nr(c_e(t), u_e(t), A_{UQ}) \leq abr_e(c_e(t))$ guarantees that the maximum achievable bit rate constraint of each transceiver is preserved.

QoS assignment. This is expressed through set $A_{UQ} = \{q_e(u) \mid u \in U_e\}$. Function $q_e(u)$ is the QoS level that will be offered to user u, by reconfigurable element e. Each function should preserve the relation $q_e(u) \in Q(s(u), u)$, for ensuring compliance with the profiles and agreements, i.e., the provision of applications at the appropriate, acceptable QoS levels.

Objective function. The reconfiguration decisions should optimise an objective function that consists of two main parts. (i) The first part is targeted to the maximisation of the aggregate utility volume deriving from the A_{UQ} allocation. This is the quantity $\sum_{u \in U_e} uv[s(u), q_e(u), u]$. The rationale is that users should be assigned to their most preferred QoS levels, to the largest extent possible. (ii) The second part of the objective function is targeted to the minimisation of the number of required changes. These changes are seen as the cost of reconfiguring the element, according to allocation A_{TC}. The rationale is that among reconfigurations that exhibit the same performance, those that require fewer changes should be preferred. The number of changes can be linked with the RAT and spectrum changes in each transceiver (e.g., see Figure 2(a)).

4. ROBUST DISCOVERY OF RECONFIGURATION CAPABILITIES

This section presents the learning and adaptation method for robustly estimating the probability that reconfiguration c is associated with achievable bit rate $abr_e(c)$ and coverage $cov_e(c)$. Let it be noted that the term 'achievable bit rate' is used to denote the reconfiguration's achievable throughput (in Kbps or Mbps) and not the traffic load (number of active sessions).

4.1. Formulation through Bayesian networks

Figure 4 (a) depicts the Bayesian network that is proposed for modeling the specified problem. ABR and COV are random variables representing the achievable bit rate and coverage, respectively. CFG is another random variable representing a configuration, e.g., we can have $CFG = c = (r, f)$. CFG is the Bayesian network's predictive attribute (node), while ABR and COV are the target attributes. The goal is the computation of the maximum value of the joint conditional probability $\Pr\left[ABR, COV \middle| CFG\right]$. Thus, given a configuration *CFG=c*, we search for the values of *ABR* and *COV* that maximize the aforementioned joint conditional probability. These values constitute the robustly estimated (i.e., the most probable) values of *ABR* and *COV* for this configuration. With reference to the model of Figure 4 (a), the desired probability $\Pr\left[ABR, COV \middle| CFG\right]$ is equivalent to the product of the conditional probabilities $\Pr\left[ABR \middle| CFG\right]$ and $\Pr\left[COV \middle| CFG\right]$, i.e:

$$\Pr\left[ABR, COV \middle| CFG\right] = \Pr\left[ABR \middle| CFG\right] \cdot \Pr\left[ABR \middle| CFG\right] \qquad (1)$$

Hence, for performing the computations, two independent conditional probability tables (CPTs) can be organized, one for *ABR* and one for *COV*. Figure 4 (b) shows a sample of the structure of a CPT for *ABR*. The CPT for *COV* is similar. Each CPT refers to a particular RAT. Therefore, $2\cdot\left|R_e\right|$ CPTs are required for the full information (*ABR* and *COV*). Each column of the CPT refers to a specific configuration (i.e., RAT and carrier frequency). Each line of the CPT corresponds to an *ABR* value, i.e. a discrete set of potential *ABR* values has been defined. Each cell (intersection of line and column) provides the probability that the configuration (corresponding to the column) will achieve the potential *ABR* value (corresponding to the line). Given a configuration, the most probable value of *ABR* is the value that corresponds to the maximum conditional probability.

Figure 4 (b) is an example for an arbitrary RAT r_1. There are $\left|F_{r_1}\right|$ columns, corresponding to configurations c_1, ... , $c_{\left|F_{r_1}\right|}$, and m lines, corresponding to *ABR* values abr_1, abr_2,..., abr_m. Without loss of generality, enumeration is done in ascending order, i.e. $abr_1 < abr_2 < \ldots < abr_m$). In other words, abr_m is the maximum value. The cell at the intersection of line i and column j is a probability value. It expresses the probability that achievable bit rate abr_i will be offered, given the fact that configuration c_j has been selected. Formally, this is denoted as $\Pr\left[ABR = abr_i \middle| CFG = c_j\right]$.

4.2. Solution: learning and adaptation

In the previous subsection, we defined that the capabilities of configurations are modeled through the CPTs. The next step is to describe how to update the CPTs. Figure 4 (c) illustrates this learning and adaptation process, which yields the robust methods for discovering the performance capabilities of candidate configurations. We focus on *ABR*, since the analysis for *COV* remains the same.

The process takes into account various measurements of the system and, more specifically, the "distance" (absolute difference) between each candidate value and the measured value. Let us assume that measurements (obtained through basic discovery-sensing functionality) show that a specific configuration can achieve bit rate abr_{meas}. This value can be updated by simple or sophisticated models that take into account measurements related to the faced interference, pathloss etc. Our management functionality can operate by exploiting minimal information. Measurements can be mapped to the achievable bit rate. In this respect, abr_{meas} values are used for fine tuning the values of the CPTs, so as to increase the degree of assurance of future predictions. Let $dif_{\max}$ be the maximum difference between the candidate *ABR* values, i.e. $dif_{\max} = abr_m - abr_1$.

Then, the following correction factor, cor_i, can be computed for each candidate *ABR* value abr_i:

$$cor_i = 1 - \frac{\left|abr_i - abr_{meas}\right|}{dif_{\max}} \qquad (2)$$

It holds that $0 \le cor_i \le 1$. A value close to 1 reflects that the corresponding *ABR* value abr_i is close to the measured value abr_{meas}, thus it should be reinforced accordingly. The opposite stands for a value that is close to 0.

The correction of the $\Pr\left[ABR = abr_i \middle| CFG\right]$ values can then be done as follows for each candidate *ABR* value abr_i:

$$\Pr\left[ABR = abr_i \middle| CFG\right]_{new} = nf \cdot cor_i \cdot \Pr\left[ABR = abr_i \middle| CFG\right]_{old} \qquad (3)$$

The parameter nf is a normalizing constant whose value can be computed by requiring all the "new" probabilities to sum up to 1.

The system *converges* when the most probable candidate *ABR* value (i.e. the one with the maximum probability) is reinforced, while the probabilities of the other candidate *ABR* values are either reduced or reinforced less. After convergence, we limit the number of consecutive updates that can be done on the probability values associated with each *ABR* value. This is done for assisting fast adaptation to new conditions. For the same reason, we do not allow that a probability falls under a certain threshold, a/m, where $0< a <1$ (m is the number of potential *ABR* values). In such cases, the normalization factor, nf, is computed by requiring all the other "new" probabilities to sum up to 1-($k \cdot a/m$), where k is the number of probabilities that are assigned equal to the threshold.

5. Selection of Reconfigurations

This section presents the second part of the solution of the DCLR problem. It exploits the potential capabilities of candidate reconfigurations, which are learned from the "robust discovery" described in the previous section. This yields a rating of reconfigurations, and eventually, leads to the best reconfigurations.

5.1. Overall solution method

In general, this part of the solution of the DCLR problem consists of four phases (Figure 5).

The first phase finds different valid transceiver reconfigurations. Each A_{TC} configuration leads to the launch of a sub-problem. The sub-problems obtained can then be processed in parallel.

In the second phase, in each of the sub-problems (with fixed A_{TC} allocation), the demand distribution, A_{TU}, is computed. At this phase, the QoS levels offered to users are kept to their lowest acceptable values (A_{UQ}^{basic}). If a solution (allocation A_{TU}) cannot be provided under these conditions, the reconfiguration is rejected.

Then, in the third phase, the QoS level offered to users is gradually augmented, until either no further increase is possible (users are at the maximum QoS level) or no higher bit rate can be offered through the examined configuration. This phase gives the final A_{UQ} allocation.

Essentially, the previous phases provide a rating of reconfigurations, with respect to the objective function. So the last, or fourth, phase selects the best reconfiguration, i.e., the reconfiguration with the highest objective function value.

The phases of the solution method described briefly above, are more thoroughly presented in the next subsections.

5.2. First phase: computation of candidate A_{TC} configurations

This phase finds the candidate reconfigurations, which will be processed in phase two and three, through dedicated sub-problems (which can be processed in parallel). A simple algorithm is to enumerate the reconfigurations, disqualifying those that are inconsistent with the operator policies or infeasible from the operational point of view. For example, the assignment of all the transceivers to a WLAN access technology, or the assignment of certain spectrum carriers to a RAT, may conflict with agreed upon NO operational rules. Likewise, the use of adjacent ("close enough") spectrum carriers in the same element, will typically not result in the optimal performance

There is an upper bound to the number of sub-problems that will be launched. Taking $|F_e|$ as the maximum number of frequencies that can potentially be assigned to any transceiver $t \in T_e$, the upper bound on the complexity is $|F_e|$!/($|T_e|$!)·($|F_e|$!- $|T_e|$!). This quantity represents the number of possible combinations, for the placement of $|T_e|$ same objects in $|F_e|$ different boxes, where each box may contain no more than one objects. In practice, the number of sub-problems launched will be significantly smaller, as some configurations will be disqualified.

5.3. Second phase: Allocation A_{TU} with A_{UQ}^{basic}

The input in this phase consists of: (i) the service area requirements extracted from monitoring information, (ii) the element and user profiles and NOs' agreements, and (iii) the reconfiguration $A_{TC} = \{ c_e(t) \mid t \in T_e \}$ selected from the previous phase. The A_{TC} allocation, in conjunction with the discovery information, yields the anticipated achievable bit rate, $abr_e(c_e(t))$, and coverage pattern, $cov_e(c_e(t))$, of each transceiver. These capabilities have to be exploited, in the distribution of users to transceivers.

Initially, the algorithm exploits the coverage information. Each user can be served by (falls within the range of) a number of candidate transceivers. This number ranges from 0 to $|T_e|$. Users out of coverage are appended to the set of "overflow" users. The rest of the users are ordered with respect to the number of candidate transceivers (starting from those with one, and continuing with those with more).

Next, the algorithm exploits the maximum bit rate information. The algorithm evolves in phases. Each phase is targeted to a particular user. The algorithm favours the assignment of a user to the transceiver that has adequate capacity in terms of bit rates offered to users, high remaining capacity (after the assignment) and low potential demand (to be faced in the future). Potential demand comes from users still not distributed that fall in the range of the transceiver. The user is appended to the set of "overflow" users if no transceiver has adequate capacity. The algorithm terminates successfully when all users are assigned to transceivers, and the volume of "overflow" users (is not large, and therefore) can be served through cooperating networks [7,8].

The formal description of the algorithm requires the definition of some supplementary data structures. The coverage information is described as follows: each user u can be served by a set of *candidate transceivers* $Ct(u) = \{ t \in T_e \mid l(u) \in cov_e(c_e(t)) \}$. Set $Ct(u) \subseteq T_e$ includes transceivers that have user u in their coverage range. The $Ct(u)$ sets readily give the set of *candidate users*, $Cu(t)$ ($Cu(t) \subseteq U_e$), of each transceiver t. Based on these definitions, we can have, for each transceiver $t \in T_e$, the potential demand, $pd(t) = nr(c_e(t), cu(t), A_{UQ}^{basic})$, and the remaining capacity, defined as $rc(t) = abr_e(c_e(t)) - nr(c_e(t), cu(t), A_{UQ}^{basic})$. Finally, the set of "overflow" users is O_e.

The algorithm evolves as follows.

Step 1. The sets $Ct(u)$ and $Cu(t)$ are computed, for all $u \in U_e$ and $t \in T_e$. The values $pd(t)$ and $rc(t)$ are initialised, for all $t \in T_e$. The users that have $|Ct(u)|$ =0 are appended to the O_e set. If $|O_e|$ exceeds a certain threshold, which means that this demand cannot be served through network cooperation [7,8], the reconfiguration A_{TC} is directly rejected, due to coverage reasons, and a transition to step 5 occurs.

Step 2. Users that have $|Ct(u)|$ ≠0 are appended to a list, L. Users are ordered users in ascending order of the $|Ct(u)|$ value (at the beginning there are those that have lower values, at the end there are those with higher values).

Step 3. The first user, u, is extracted from L. User u is assigned to the transceiver t of $Ct(u)$ that has sufficient remaining capacity, $rc(t)$, as well as the lowest value of the heuristic measure $pd(t)/rc(t)$. In this way, the algorithm favours the transceiver that has low potential demand and high remaining capacity. If such a transceiver is found, u is appended to $u_e(t)$, otherwise, u is appended to O_e. If $|O_e|$ exceeds a certain threshold, the reconfiguration is rejected (due to insufficient capacity and coverage).

Step 4. If L is not empty, there are more users unassigned. Therefore, the following actions are conducted: (i) the $rc(t)$ value, of the transceiver that was selected in the previous step, is updated; (ii) for all transceivers in the $Ct(u)$ set, the $Cu(t)$ sets and the $pd(t)$ values are updated, by removing the contribution of user u; (iii) the $Ct(u)$ set is discarded; (iv) a transition to step 3 is performed.

Step 5. End

5.4. Third phase: Allocation A_{UQ} final

The input in this phase consists of (i) the service area requirements (monitoring information), (ii) the profiles and agreements, and (iii) the allocations A_{TC} and A_{TU} selected from the previous phases. The goal is to maximise, if possible, the QoS levels that will be offered to users, because of the reconfiguration.

Depending on the profile, the improvement of the QoS level of a user, u, may raise the corresponding utility volume, by a factor $\delta uv(u)$. This will enhance the overall objective function value. Therefore, we can obtain a set of candidate moves (QoS improvements), which can be ordered with respect to their potential positive contribution to the objective function value. In each phase, the algorithm selects the most efficient improvement (i.e., the one that improves most the objective function), which is also feasible from the transceiver maximum bit rate point of view. The algorithm stops when no more QoS improvement can be made, or when the transceivers do not have remaining capacity and thus cannot support higher bit rates.

The algorithm evolves as follows.

Step 1. The algorithm computes the $\delta uv(u)$ values, of each user $u \in U_e$, which will derive from the improvement of the current QoS level by one level. If $\delta uv(u)=0$ for all $u \in U_e$, i.e., QoS cannot be improved, the algorithm ends (transition to step 6).

Step 2. The users that have $\delta uv(u)>0$ are appended to a list, L. Users are ordered in descending order of their $\delta uv(u)$ value.

Step 3. If $L \neq \emptyset$, the first user u is extracted from the list. Otherwise, the algorithm ends (transition to step 6).

Step 4. User u will be assigned to higher QoS (e.g., $q_e(u)=q_e(u)+1$) if the improvement is feasible from the transceiver capacity (maximum supported bit rate) point of view. If the improvement is not feasible, user u will not be further considered and a transition back to step 3 occurs.

Step 5. The algorithm investigates the impact from further improving the QoS level of u. The new, $\delta uv(u)$, is computed. Then, u is appended to the appropriate place of L, and a transition back to step 3 is conducted.

Step 6. End

5.5. Fourth phase: Selection of best reconfiguration

This last phase is targeted to the selection of the best reconfiguration. The selected reconfiguration optimises the objective function value, in the sense that it results in maximal aggregate utility volume, and in minimal changes in the transceivers.

6. RESULTS

This section presents results on the efficiency of the robust discovery and reconfiguration selection methods.

6.1. Robust discovery

Three scenarios are realized. The scenarios are driven by the estimation of *ABR's* conditional probability values. The coverage values can be determined in the same way. In all scenarios, parameter a has been set equal to 0.1. Moreover, the subsequent definitions apply: the term *scenario case* refers to the execution of the same scenario with the sole difference of modifying the value of one of the scenario's parameters. Thus, in scenario 1, four cases are studied, corresponding to four different values of parameter abr_{meas}, while, in scenario 2, two cases are studied, with regard to two different values of the allowed number of consecutive reinforcements. The term *scenario phase* is used to distinguish temporal parts of the same scenario in which the value of abr_{meas} changes. Thus, in scenario 3, three phases are observed, since three different abr_{meas} values are sensed as time passes. Finally, the time epochs in which an abr_{meas} measurement is taken are called *scenario steps*. Scenario steps within the same scenario phase are characterized by the same measured abr_{meas} value.

Scenario 1. This scenario helps the in-depth understanding of our proposed technique. Our focus is on an arbitrary configuration c_1 =(r_1 , f_1). It is assumed that there are m =5 candidate *ABR* values (in Mbps): abr_1 =0.5, abr_2 =1.0, abr_3 =1.5, abr_4 =2.0, abr_5 =2.5. Hence, $dif_{\max}$ =2 Mbps. It should be noted that a denser grid of candidate values could be selected (actually, in this case our results would have been favored). What is more, the distance between two subsequent candidate values needs not be the same. Also, only three consecutive reinforcements are allowed, after convergence.

Figure 6 (a)-(d) depicts the distribution of conditional probabilities in four cases. In each case abr_{meas} is (in Mbps) 1.2, 1.5, 2.1 and 0.75, respectively. The algorithm is applied in five series of runs. Initially, the conditional probabilities are uniformly distributed, i.e. equal to 0.2, in all four scenarios (step 1). By using (6), we calculate the correction factors. Then, by using (7), we compute the new (adjusted) conditional probabilities. The results for each case are further analyzed in the following.

Figure 6 (a) shows that our model correctly and quickly adapts to the situation, by selecting abr_2 as the most probable value, in the second step. As expected in this case, from the beginning there are high values for abr_2 and abr_3 , a slight diminishment for abr_1 and

abr_4 , and a severe degradation for abr_5 . As the scheme is further applied, and since abr_{meas} does not change, the most probable value is actually further reinforced. Our learning and adaptation model accurately adapts to the second case also, in which abr_{meas} =1.5 (Figure 6 (b)). As can be observed, abr_3 quickly prevails, whereas abr_2 and abr_4 remain practically the same, and finally, abr_1 and abr_5 suffer significant diminishment. Figure 6 (c) shows the results from the third case, in which abr_{meas} =1.8. The model quickly adapts to abr_4 , with the probabilities of abr_5 and abr_3 coming next, and those of abr_1 and abr_2 significantly falling.. The model is also robust in the last, rather unlikely, case, in which abr_{meas} =0.75 (Figure 6 (d)). The model suggests abr_1 and abr_2 as the most likely values. The probability of abr_3 is slightly increased, while the probabilities of abr_4 and abr_5 are degraded.

Scenario 2. The goal of this scenario is mainly to examine how many steps it takes for the scheme to adapt to a radically new situation, in other words, to a sudden and significant change in the environment conditions. Figure 7 (a)-(b) shows the speed of the adaptation when there is a sudden degradation of the measured achievable bit rate. In the first case, we allow three consecutive reinforcements of the most probable value, after convergence. In the second case, only one reinforcement is allowed..

Figure 7 (a) shows what happens when abr_{meas} suddenly becomes 1.1 Mbps. Our starting point is the case depicted in Figure 6 (c) (i.e., our model evolved as depicted in Figure 6 (c) before the sudden change in abr_{meas}). The goal is to examine how quickly the system can adapt and converge to abr_2 , which is the most proper value, based on the measurements. As can be observed in Figure 7 (a), in 4 steps (steps 2-5) the most probable value drops from abr_4 to abr_3 . In another six steps (steps 6-11), candidate value abr_2 and abr_3 are suggested as the most likely ones. Finally, in the next step (step 12) abr_2 becomes the most probable one.

Figure 7 (b) shows what happens if only one reinforcement is allowed after convergence. Again, Figure 6 (c) is our starting point and abr_{meas} is 1.1 Mbps. In just 2 steps (steps 2-3) the most probable value drops from abr_4 to abr_3 . In another 3 steps (steps 4-6), candidate value abr_2 almost reaches abr_3 , and in the next step (step 7) abr_2 becomes the most probable one.

An important conclusion that can be deduced from the aforementioned simulations is that the number of consecutive reinforcements after convergence clearly affects the model's adaptation speed. A high *number of consecutive reinforcements* reduces the *adaptation speed*, i.e. the *convergence rate*. As an imminent result of this, a low convergence rate renders the system more immune to temporary changes of the sensed metrics (abr_{meas} value). The trade-off here is that if the sensed change is both permanent and substantial (from a quantitative perspective), then the system will delay more in adapting to the new situation (as happened in the case of Figure 7 (a) compared to that of Figure 7 (b))..

Scenario 3. Our focus is on a second arbitrary configuration $c_2 = (r_2, f_2)$. The goal of this subsection is to provide a more detailed and complex scenario, so as to demonstrate both the efficiency and the adaptation speed of the proposed probabilistic model. Especially regarding the model's adaptation speed, it should be noted that the scenario presented below can be classified as one of the worst case scenarios, due to the radical changes of the measured values.

It is assumed that there are 11 candidate values (Mbps): abr_1 =1, abr_2 =2, abr_3 =3, … , abr_{11} =11. Then, $dif_{\max}$ =10 Mbps. Again it is assumed that only three consecutive reinforcements are allowed after convergence. The scenario consists of three phases described in the following. Figure 8 (a)-(c) depicts the distribution of conditional probabilities for each phase. During the first phase, the value of abr_{meas} is 7.3 Mbps, in the second phase it increases to 9.8 Mbps, while in the third phase it severely falls down to 3.8 Mbps.

Figure 8 (a) reflects the behavior when abr_{meas} =7.3 Mbps. At first, the conditional probabilities of the candidate values are uniformly distributed, as depicted in step 1. The robust discovery functionality correctly adjusts (step 2), by selecting abr_7 as the most probable value. During the next steps (steps 3-5), the measured value does not change, so the most probable value remains the same, and is actually further reinforced. As has been explained, the number of allowed further reinforcements depends on the implementation. In this scenario, three further consecutive reinforcements (steps 3-5), after convergence, are allowed. Higher values assist the system in avoiding oscillations, while lower values help it adjust more quickly to severe and relatively permanent changes.

Figure 8 (b) shows what happens when, at some time, the measured value changes to abr_{meas} =9.8 Mbps. The last phase of Figure 8 (a) is the starting point for this phase, i.e., step 1 of Figure 8 (b) is the same as step 5 of Figure 8 (a). As depicted in Figure 8 (b), the model's performance is quite satisfactory, as in only 2 steps (step 3) the most probable value shifts from abr_7 to abr_8. In another 3 steps (step 6), the model selects abr_9 as the most probable value, and, finally, in another 4 steps (step 10), abr_{10} (which is the nearest to the measured value) is selected. Thus, in this phase, the system proves to be both fast in adaptation, and also careful (the entire adaptation process takes more than just 1 or 2 steps), in order to avoid oscillations. It is also worth noting that, as may be observed in Figure 8 (b), the conditional probabilities of abr_1 (steps 2-10), abr_2 (steps 2-10), abr_3 (steps 3-10), abr_4 (steps 4-10) and abr_5 (steps 7-10) have been set by the model to the minimum allowed value (i.e., the threshold's value). This can help the model in future adaptations, as in the case of the next phase of the scenario.

In Figure 8 (c), the system is challenged to adapt to an extremely radical change. The *ABR*'s measured value, abr_{meas}, drops from 9.8 to 3.8. Figure 8 (c) depicts how the model adapts to the situation. Step 1 represents the distribution of the final step of Figure 8 (b). As may be observed, in just 1 step (step 2) the most probable value drops from abr_{10} to abr_9. In another 3 steps (step 5), the most probable value shifts from abr_9 to abr_8. In the next step (step 6), candidate value abr_4 (which is the closest to the measured value), as well as the candidate values in the neighborhood of abr_4 increase significantly. In the next step (step 7), value abr_4 outperforms the others. Thus, the model managed to adapt in only 6 steps, which is a remarkable performance. The "threshold" mechanism has also contributed in the model's satisfactory adaptation speed, by not allowing the conditional probability of abr_4 and its neighbors to drop below an appropriate minimum value.

6.2. Reconfiguration selection - exploitation of reconfiguration capabilities

Four scenarios are realized. Initially, we present the input given to the scheme. Then, we analyse the results from the scenarios.

Monitoring Information. Figure 9 (a) shows a simple service area that is covered by a network segment. The segment consists of a number of reconfigurable elements, as the one shown on Figure 9 (b). Elements operate in parallel. The behaviour of these elements and the service area requirements cause reconfiguration triggers to the shaded element in the figure, which will be in the focus of this subsection. Figure 9 (c) refers to the demand in the element's service area. Nine different cases studied. Each case corresponds to a different traffic mix (combination of voice and data sessions). Initially, the demand for voice dominates. Gradually, the demand for the data service dominates. The demand is taken uniformly distributed within the service area.

Profiles. Each element is equipped with 3 reconfigurable transceivers (Figure 9 (b)). Each transceiver may select between the two configurations studied in the previous sub-section. In doing so, the resulting overall configurations for each element can be denoted as e.g. (c_1, c_1, c_2), implying that two transceivers are assigned configuration c_1, while the third one is assigned configuration c_2 and so forth. Additionally, the assignment of configuration c_2 to all transceivers is not considered, since it would lead to coverage holes.

Set S_e consists of two services, a voice service (s1) and a data service (s2). Whereas the voice service is associated with a fixed quality level, for the data service, a set of quality levels is provided. Moreover, s1 can only be offered through r_1 (therefore, configuration c_1). Figure 9 (d) contains the acceptable QoS levels, the utility volume, whenever a service is offered at a certain QoS level, as well as the bandwidth requirements per service.

Discovery Procedures. As aforementioned, the configurations c_1 and c_2 have different bit rate capabilities, 1 or 2Mbps for c_1, and 4, 7 or 10Mbps for c_2. It is also assumed that c_1 can achieve larger coverage than c_2, i.e., the larger the supported bit rate is, the smaller the coverage becomes.

All in all, we are able to consider various scenarios, combining the capabilities (maximum bit rate and coverage) of the configurations, in order to see which configuration fits better the traffic mixes (by applying the test cases shown in Figure 9 (c)).

Scenario 1. This scenario assumes that $abr_e(c_1)$ =1 Mbps and $abr_e(c_2)$ =4 Mbps. The coverage pattern for c_1 is about 1000m, and for c_2 about 500m. Figure 10 (a) shows indicative results.

Configuration (c_1, c_1, c_1) increases the objective function value as the data load increases. This happens because, at the same time, voice load decreases, and therefore, there is spare remaining capacity that can be exploited in offering higher QoS to more data sessions. At some point, the objective function value remains the same (cases 5-8), since the increase deriving from new data sessions is compensated by the decrease in the voice sessions. Finally, in case 9, the data sessions have become so many, that for some users the QoS levels offered need to be degraded, compared to case 8, and thus, the objective function value decreases.

The behaviour of the (c_1 , c_1 , c_2) configuration is similar. Initially (cases 1-3), the configuration cannot handle the demand, because the voice load dominates and exceeds the achievable bit rate capabilities of the two c_1 transceivers. Starting from case 4, the voice sessions have decreased and can be accommodated by the two transceivers, configured with c_1. Consequently, (c_1, c_1, c_2) yields the highest objective function value. This occurs since the spare remaining capacity is exploited for providing higher QoS to data services. Higher objective function values are achieved, compared to (c_1, c_1, c_1), because $cp_e(c_2)$ is higher.

At some point the improvement stops, because the overall load is heavy, and therefore, some of the QoS levels have to be degraded again. Figure 10 (a), diagram in the right, shows the corresponding QoS levels offered for (c_1, c_1, c_2), indicatively.

Finally, configuration (c_1, c_2, c_2) exhibits an acceptable performance only at certain traffic mixes. Specifically, its objective function value is initially zero, until the voice sessions can be accommodated by a single c_1 transceiver. This occurs in case 7. Then, (c_1, c_2, c_2) proves itself to be appropriate, but only until the data sessions have become far too many and cannot be catered for by c_2's limited coverage (the distribution of users within the element is uniform).

Comparing now the alternatives, we find that at the very initial demand patterns, the (c_1, c_1, c_1) configuration performs better. However, as data sessions increase, the (c_1, c_1, c_2) configuration becomes superior, due to the spare remaining capacity that can upgrade QoS levels offered to continuously coming data sessions. This excellence of (c_1, c_1, c_2) is though sometimes marginal compared to (c_1, c_1, c_1). Additionally, at certain traffic mixes with few voice and many data sessions, (c_1, c_2, c_2) exhibits the best performance, due to its potential to offer higher bit rates.

Scenario 2. This scenario assumes that $abr_e(c_1)$ =1 Mbps and $abr_e(c_2)$ =7 Mbps. The coverage pattern for c_1 is about 1000m, and for c_2 about 250m. Figure 10 (b) shows indicative results.

The study of the curve on the left, leads to conclusions that are similar to the ones obtained in scenario 1. Regarding configuration (c_1, c_1, c_1), exactly the same results are obtained, as expected, since $abr_e(c_1)$ =1. Regarding the (c_1, c_1, c_2) configuration, initially, it is inappropriate. Starting from case 4, its performance increases, along with the increase in the data sessions, up to the point where the system reaches its "pole" bit rate point. However, it should be noted that this configuration gives now higher objective function values than scenario 1, due to the fact that $abr_e(c_2)$ = 7 Mbps. Again, configuration (c_1, c_2, c_2) exhibits no performance at the initial traffic loads, and is, in general, appropriate only when voice sessions have decreased enough and can be served by a single c_1 transceiver. Moreover, (c_1, c_2, c_2) provides now a higher value in case 7 than in scenario 1 (due to the current higher $abr_e(c_2)$ value), but proves itself inappropriate right after that, due to its restricted coverage capabilities that cannot cater for the uniformly distributed coming data sessions.

In general, comparing again the available configurations, we find that at the very initial demand patterns, (c_1, c_1, c_1) outperforms the rest configurations. As data sessions increase, there is superiority of the (c_1, c_1, c_2) configuration, which is important compared to (c_1, c_1, c_1). Additionally, (c_1, c_2, c_2) performs better than the rest configurations at certain traffic mixes (case 7).

Scenario 3. This scenario assumes that $abr_e(c_1)$=2 Mbps and $abr_e(c_2)$=4 Mbps. The coverage pattern for c_1 is about twice the coverage of c_2. Figure 10 (c) shows indicative results.

Examining (c_1, c_1, c_1) at a first stage, the results obtained are naturally same in rationale as in the previous scenarios. However, the fact that $abr_e(c_1)$=2 Mbps leads in general to a higher objective function value compared to the previous scenarios. Regarding (c_1, c_1, c_2), it can be put into effect also from case 3, since the larger achievable bit rate capabilities of the 2 c_1 transceivers is enough for the existing voice sessions. Finally, (c_1, c_2, c_2) still remains appropriate only in certain traffic patterns. Specifically, it could be taken into consideration only in cases 6, 7 and 8, where, on one hand, voice sessions can be efficiently served by one c_1 transceiver, whilst data sessions fall within c_2's coverage area. Much to our anticipation, this curve moves to the left in comparison to scenarios 1 and 2 (where $abr_e(c_1)$=1 Mbps), justified by the fact that now a single c_1 transceiver can accommodate more voice sessions than before.

The comparison among the configurations examined, shows again that at the very initial demand patterns, (c_1, c_1, c_1) outperforms the rest configurations. As data sessions increase, there is a considerable superiority of the (c_1, c_1, c_2) configuration, even more important than in scenarios 1 and 2, due to the large capacity of c_1. The (c_1, c_2, c_2) configuration performs better than the rest configurations at certain traffic mixes (cases 6 and 7), since it can more efficiently guarantee the desired utility.

Scenario 4. This last scenario assumes that $abr_e(c_1)$=2 Mbps and $abr_e(c_2)$=7 Mbps. The coverage pattern for c_1 is about three times the coverage of c_2. Figure 10 (d) shows indicative results.

Taking a look first at configuration (c_1, c_1, c_1), the results obtained are exactly the same as in scenario 3, since $abr_e(c_1)$ =2 Mbps. Regarding (c_1, c_1, c_2), the initial performance (obtained again only after case 3), increases as the data sessions become more (along with a total increase in the system's load), offering higher QoS levels, using both c_1 and c_2 transceivers. Finally, (c_1 , c_2 , c_2) is still only in certain traffic patterns appropriate. Specifically, it could be taken into consideration only in cases 6 and 7 in this scenario.

7. CONCLUSIONS

B3G wireless infrastructures can be efficiently realized by exploiting cognitive networking potentials. Cognitive, wireless access, infrastructures dynamically reconfigure to the appropriate RATs and spectrum, in order to adapt to the environment conditions and requirements. This is achieved by disposing reconfigurable platforms, controlled and supported by advanced management functionality that endows them with cognitive features. This paper provided such management functionality by addressing a problem, called "Distributed, Cross-Layer Reconfigurations" (DCLR). The paper (i) defined and solved a fully distributed problem version, (ii) proposed robust learning and adaptation, strategies for discovering the performance potentials of alternate reconfigurations, (iii) gave a computationally efficient solution to the problem of exploiting the performance potentials of reconfigurations, and (iv) presented results that expose the behaviour and efficiency of the developed schemes.

The solutions presented herein are based on a set of case studies. Specifically, the work reveals the advantages of cognitive wireless networks compared to the traditional ones, in terms of enabling fast, reliable and stable adaptations to the environment, based on knowledge and experience. The algorithms presented are fast, correct and dispose a good convergence rate. In addition, they are generic by nature, since they address arbitrary configurations and thus they provide a generalised framework which may be applicable to the management of various radio access technologies and frequency bands. In this respect, the proposed management scheme might form part of modules of future reconfigurable transceivers.

In the course of future activities, autonomic computing principles shall be further employed, by developing an autonomic manager, which will encompass the DCLR scheme. The manager will consist of policies, context perception capabilities, reasoning algorithms, learning functionality and knowledge engineering, technologies for the representation of ontologies and semantics. All these will yield a system that *hypothesises* on causes to a problem, and subsequently, validates or falsifies the hypothesis.

Another issue for future study is to complement the distributed DCLR scheme with a second tier of, more centralised, management functionality. The centralised functionality will be invoked when the distributed components cannot converge to acceptable solutions. The synergy of the two tiers will guarantee that whenever the distributed components diverge from the near-optimal performance levels, the application of the second tier will restore the performance to the desired levels.

Finally, subject for future research is to exploit the DCLR scheme for enabling NOs to *personalise* their service offerings, instead of limiting subscribers to a fixed set of inflexible choices. Seamless mobility applications can build on schemes like DCLR to intelligently change the services that they provide based on business policies and context.

8. ACKNOWLEDGEMENT

This work has been performed in the framework of the European Funded (EU) funded project End-to-End Reconfigurability II (E^2R II), as well as in the context of ARIADNE project (03EΔ235), partially funded by the General Secretariat of Research and Technology (GSRT) of the Greek Ministry of Development.

9. References

[1] Third (3rd) Generation Partnership Project (3GPP), Web site, www.3gpp.org , 2007

[2] Third (3rd) Generation Partnership Project – 2 (3GPP2), Web site, www.3gpp.org , 2007

[3] Institute of Electrical and Electronics Engineers (IEEE), 802 standards, Web site, www.ieee802.org, 2007

[4] Wireless World Research Forum (WWRF), www.wireless-world-research.org, 2007

[5] Digital Video Broadcasting (DVB), Web site, www.dvb.org, 2007

[6] Internet Engineering Task Force (IETF), www.ietf.org, 2007

[7] P.Demestichas, G.Vivier, G.Martinez, L.Papadopoulou, V.Stavroulaki, F.Galliano, M.Theologou, "Wireless beyond 3G: managing services and network resources", *IEEE Computer*, Vol. 35, No. 8, pp. 96-98, 2002

[8] P.Demestichas, V.Stavroulaki, L.Papadopoulou, A.Vasilakos, M.Theologou, "Service configuration and traffic distribution in composite radio environments", *IEEE Transactions on Systems, Man and Cybernetics* Journal, vol. 33, No. 4, pp. 69-81, Nov. 2003

[9] FP6/IST project E2R II (End-to-End Reconfigurability – 2nd phase), www.e2r2.motlabs.com, 2007

[10] P.Demestichas, G.Vivier, K.El-Khazen, M.Theologou, "Evolution in wireless systems management concepts: from composite radio to reconfigurability", *IEEE Commun. Mag.*, Vol. 42, No. 5, pp. 90-98, May 2004

[11] Software Defined Radio Forum (SDRF) Web site, www.sdrforum.org, 2007

[12] P. Demestichas, G. Dimitrakopoulos, K. Moessner, T. Dodgson, D. Bourse (editors – ch.8: Reconfigurability), "*Technologies for the Wireless Future: Wireless World Research Forum (WWRF)*, Vol. 2", J. Wiley and Sons, 2006, ISBN: 0-470-02905-6

[13] R. Thomas, D. Friend, L. DaSilva, A. McKenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives", *IEEE Commun. Mag*. Vol. 44, No. 12, Dec. 2006

[14] S. Haykin, "Cognitive Dynamic Systems", *Proceedings of the IEEE*, vol. 94, no. 11, Nov. 2006, pp. 1910-1911

[15] I. F. Akyildiz, W.-Y. Lee, M.C. Vuran, S. Mohanty, Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey, *Computer Networks*, vol. 50, no. 13, Sep. 2006, pp. 2127-2159

[16] J. Mitola III, G. Maguire Jr. "Cognitive radio: making software radios more personal", *IEEE Personal Commun.*, Vol. 6, No 4 , pp13 -18, August 1999

[17] P.Demestichas, G.Dimitrakopoulos, J.Strassner, D,Bourse, "Introducing Reconfigurability and Cognitive Networks Concepts in the Wireless World: Research Achievements and Challenges", *IEEE Vehicular Technology Magazine*, Vol. 1, No. 2, June 2006, pp. 32-39.

[18] E. Gelenbe, R. Lent, A. Montuori, Z. Xu, "Cognitive packet networks: QoS and performance", In Proc. IEEE MASCOTS 2002, pages 3-12, USA, Oct. 2002.

[19] R. Rubenstein, Radio Gets Smart, *IEEE Spectrum*, vol. 44, no. 4, Feb. 2007, pp. 46-50

[20] M.Gandetto, A.Cattoni, C. Regazzoni, "A distributed approach to mode identification and spectrum monitoring for cognitive radios", in Proc. Software Defined Radio (SDR' 05) Technical Conference, Anaheim, USA, November 2005

[21] R.E.Neapolitan, "*Learning Bayesian Networks – Series in Artificial Intelligence*" , Prentice Hall, 2002

[22] J. Kephart, D. Chess, "The vision of autonomic computing", *IEEE Computer*, Vol. 36, No.1, pp. 41-50, January 2003

[23] J. Strassner, *"Policy-based network management"*, Morgan Kaufmann Publishers. U.S.A., 2005

[24] J.Strassner, "Autonomics – A critical and innovative component of seamless mobility", Technical Report, http://www.motorola.com/mot/doc/5/5978_MotDoc.pdf, Motorola, Dec. 2005

[25] J.Mitola III, "Software radio architecture: a mathematical perspective", *IEEE J. on Selected Areas in Communications*, Vol. 17, No. 4, pp. 514 – 538, April 1999

[26] P.Leaves, K.Moessner, R.Tafazolli, D.Grandblaise, D.Bourse, R.Tonjes, M.Breveglieri, "Dynamic spectrum allocation in composite reconfigurable wireless networks", *IEEE Commun. Mag.*, Vol. 42, No. 5, pp. 72-81, May 2004

[27] J. Palicot, C. Roland, "A new concept for wireless reconfigurable receivers", *IEEE Commun. Mag.*, Vol. 41, No. 7, pp 124-33, July 2003

[28] M. A. L. Thathachar, P. S. Sastry, *"Networks of Learning Automata"*, Kluwer Acad. Publishers, 2004

[29] A.Mas-Colell, "*Microeconomics*", Oxford University Press, U.K., 1995

[30] J.Tirole, "*The Theory of industrial organization*", MIT Press, Cambridge, Mass., U.S.A., 1998