

A Policy-Based Framework for Autonomic Reconfiguration Management in Heterogeneous Networks

Zhao Zhenzhen
Dept. RS2M, GET-INT
9, Rue Charles Fourier
91000, Evry, France
+33 160 76 41 64

Chen Jie
Dept. Telecommunication, BUPT
10, Xitucheng Road
10086, Beijing, China

Noel Crespi
Dept. RS2M, GET-INT
9, rue Charles Fourier
91000, Evry, France
+33 160 76 46 23

zhenzhen.zhao@it.sudparis.eu

jie.chen.bupt@gmail.com

Noel.crespi@it-sudparis.eu

ABSTRACT

This paper presents a policy-based framework to approach the issue of autonomous reconfiguration management in heterogeneous networks. In contrast to existing policy-based approaches, the proposed framework addresses the management issue from a new perspective through posing it as a problem of learning from current network behavior, while creating and updating policies dynamically in response to changing reconfiguration requirements, and this task is implemented by Reinforcement Learning methodology. A two-layer policy model is used to mapping users and operators' higher level goals into network level objectives. The autonomic reconfiguration procedures for policy creation, storage, evaluation are also presented in detail. Illustrative examples analysis and simulation results demonstrate the performance of the proposed work.

Keywords

Adaptive policies, autonomic communications, policy-based management, reconfigurable systems, reinforcement learning

1. INTRODUCTION

The coexistence of heterogeneous radio access technologies (RATs) would become one of the most significant features of the future B3G environment. Various terminals and network infrastructures with reconfigurable capabilities will provide operational choices for users, service providers and operators. In order to accomplish flexible service offering and to cope with such complex systems, the need for end-to-end reconfigurable architectures, systems, and function rises [1]. Accordingly, the EU FP6 Integrated Project IST-E²R (End-to-End Reconfiguration) [1] has been set up, aiming at developing the architecture of the reconfigurable devices and the supportive system functions.

In accordance with the developing trend of autonomic communications, which is becoming more and more important as the system complexity grows higher with the increasing technologies and devices that overwhelm users and operators, the execution of reconfiguration procedures should depend on the human intervention as little as possible. One promising solution is policy-based management (PBM) [3][5]. The importance of PBM has already been shown in previous work [3-9] that it simplifies

the management procedures by establishing policies to deal with problems that are likely to occur, such as network resources allocation, quality of service (QoS) improvement and security (firewalls) guarantee. Therefore, when facing the autonomic reconfiguration problem in future communication systems, we propose a policy-based approach to solve it. However, existing management architectures [3-8] could be less suitable for a reconfigurable environment where the proposed management metrics may differ greatly across the heterogeneous RATs.

Since the reconfiguration operations should depend on the human intervention as little as possible, the policies are required to be created and modified dynamically. The competent policy management mechanisms should be able to create new policies in response to the changing requirements according to the different radio environments in a timely manner, and this makes a great challenge to the policy-based management tools. Since network-level policies are dynamically derived from business objectives and user requirements, policies have to be self-managed to adapt to these changing objectives and requirements without much human-intervened planning. Such intelligence requires a learning process to be able to reach the optimal policy from its online operations, which falls rightly within the field of reinforcement learning (RL). As a powerful tool for studying the principle of learning to act without knowing the environment model, RL has been successfully applied in different research areas such as mobile communication systems [10-13].

In this paper, we develop a new paradigm to approach the issue of autonomous policy-based management of reconfigurable communication systems. Our main contributions are two-fold. Firstly, a two-layer policy model is used to facilitate the mapping of higher-level abstract user/operator policies into network-level objectives. By decoupling the functionality of adapting network-level policies from the task of mapping business objectives and abstract user requirements, the proposed work offers users and operators the freedom to specify and dynamically change their requirements. Secondly, the policy adaptation approach, which includes policy registration, policy dispatch and policy reassessment, is proposed following the autonomic "trial-and-error" learning process. By applying the RL methodology, sets of policies are dynamically created and modified to adapt to the current resources availability and user's demands. Specifically, we adopt Q-learning [14] as the RL mechanism for its simplicity and effectiveness. In addition, an illustrative example is introduced to demonstrate the feasibility of the proposed framework.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The remainder of the paper is organized as follows. In Section 2, related work and existing approaches for policy adaptation are briefly discussed. Section 3 presents an overview of the proposed framework for autonomous reconfiguration management. Section 4 and 5 discuss the details of the proposed two-layer policy model and the runtime policy adaptation process, respectively. An illustrative example is proposed in Section 6. Performance evaluation and simulation results are discussed in Section 7. Finally, Section 8 concludes the paper.

2. RELATED WORK AND MOTIVATION

PBM has been introduced as a promising solution to the problem of managing networks. As the development of wireless communications, reconfigurable system will provide ubiquitous access and pervasive service. The reconfiguration procedures should also be managed using policy-based approaches. However, existing management approaches have nothing to do with reconfigurability, which paves the way towards new paradigm as well as challenges for the reconfiguration management.

In [3], a management architecture for mobile ad-hoc networks is proposed based on the IETF PCIM policy model [4]. This model did not take into account the context information that may improve policy decisions in different network environments. Similarly, in [6], a network management model for a multimedia service in the broadband wireless access network is presented. However, it may impose a heavy burden on the operators in analyzing management information for the policy adaptation.

Generally, PBM faces two important problems: policy refinement and policy adaptation. Without exceptions, reconfiguration management encounters the same problems.

Policy refinement is the process of transforming a high-level, abstract policy into a low-level, concrete one. Transforming business, user and administrative objectives into network element specific policies can be regarded as an example of such process. A goal-based approach for policy refinement is described in [9]. The system uses event calculus in conjunction with adductive reasoning to derive the sequence of events to achieve the desired goal. However, the system is not suitable for autonomic reconfigurable systems since it depends on full system knowledge and lacks approaches to deal with incomplete or conflicting data.

Another problem for PBM is policy adaptation. Although it seems to provide a more promising solution for network management, existing adaptation mechanisms still have certain limitations. These mechanisms usually lack an essential degree of flexibility to build upon past experiences gained from the impact of

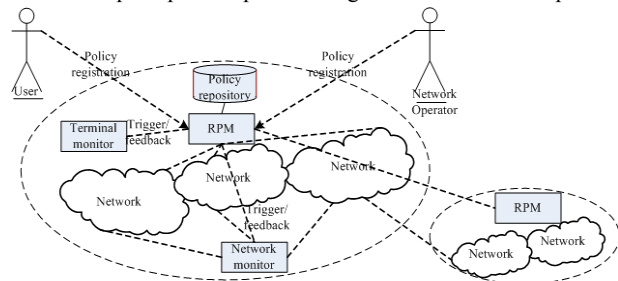


Figure 1. Proposed reconfiguration management framework.

previously pursued adaptation strategies on the current network behavior. Various research trends, e.g., [7], have highlighted the notion of policy adaptation and the central role that it can play in policy-enabled network management. This notion of policy adaptation is becoming even more crucial as the managed systems become more complicated. In [5], an autonomous policy-based management for QoS control in wired/wireless differentiated communication systems has been introduced. Our work follows the same concept of adaptation based on learning while giving more flexibility through the use of policies at different layers to allow users to dynamically specify their requirements in terms of high-level policies.

3. PROPOSED FRAMEWORK

This section presents a brief overview of the proposed policy adaptation framework. The primary goal of the framework is to reduce the complexity of autonomic reconfiguration management. More specifically, the framework should depend on the platform of E2R systems and depend on human intervention as little as possible. Furthermore, the framework should support open information model formats, i.e. Common Information Model (CIM) [15], thus users/operators-specific functions or objectives can be defined extensible and flexible.

We assume that the underlying communication system consists of a set of wireless domains that support the end-to-end reconfigurability. Figure 1 presents the schematic description of the main components of the proposed framework. The central component in this framework is the reconfiguration policy manager (RPM). The key feature in RPM design is decoupling the tasks of managing reconfiguration in policies from the functions of executing the network elements' reconfiguration.

The first task of policy-based reconfiguration management is achieved through a two-layer policy model. At the top level of this model, users and operators, respectively, specify their higher-level policies through a graphical user interface attached to the RPM. Then the higher-level objectives are refined to a set of lower-level policies. The final results of this process are a set of network-level objectives. The latter task is achieved by dispatching the appropriate policies, which is realized through an E2R defined autonomic reconfiguration process [16].

The functions of the network/terminal monitors are two-fold. On the one hand, by monitoring the network/terminal context information, they can trigger the RPM to dispatch the appropriate policies. On the other hand, they feed back the policies execution results to the RPM for updating or creating the policies. Note that before dispatching the policies, the RPM may negotiate with neighboring RPM domain to ensure the most appropriate policies are selected. Finally, the policy repository is used to store the registered policies and is accessed by the RPM.

The adaptation process is either triggered periodically in order to slightly adjust existing policies for better performance or triggered through events received from network/terminal monitors. Examples of the events are low battery level, movement of users, or changing in an application QoS requirements, etc. The subsequent two sections explain the idea of the two-layer policy model and describe the steps of the policy adaptation process, respectively.

4. PROPOSED POLICY HIERARCHY

This section describes the methodology of translating abstract higher-lever users' preferences and operators' business goals for reconfiguration into network-level objectives.

Due to the dynamic attribute of the complexity of the heterogeneous environment, it should be noted that the refinement is inconvenience when having the user define his preferences in a static manner. In the proposed work, the key solution to circumvent this problem is achieved through incorporating a hierarchical approach that enables users/operators to specify their high-level goals in terms of policy structures. Figure 2 depicts a schematic illustration for the proposed hierarchy. The hierarchy consists of two layers.

In the first layer, network operators specify higher-level goals in the form of business objectives through a graphical user interface attached to the RPM. In the same manner, users are also allowed to specify their goals based on their requirements, which are related to different parameters. For example, the business goals for each operator are to improve the resource utility and revenue, while the user preferences maybe the QoS requirement, the cost of services, or the location of users. Policy conditions are translated into events that the RPM registers with the user/operator and network monitors.

In the second layer, actions are translated into network-level objectives. Examples of actions are admission control (AC), vertical handover (VHO), resource allocation and spectrum management. In order to reach these targets in a more practical way, network-level objectives are translated into network element level policies, i.e. a set of (condition, action) pairs.

As shown in Figure 2, the two boxes are used to show the separation between the functionalities of mapping users, operators' policies into network-level objectives and the task of adapting network-level policies, the purpose of the differentiation between these two layers is to further facilitate the automation of the mapping process. An illustration of possible QoS mapping methodologies can be found in [17-19].

5. PROPOSED POLICY ADAPTATION APPROACH

In this section, we discuss the adaptation of network policies either to satisfy new users' preferences and operators' business goals or to respond the feedback information reported back by

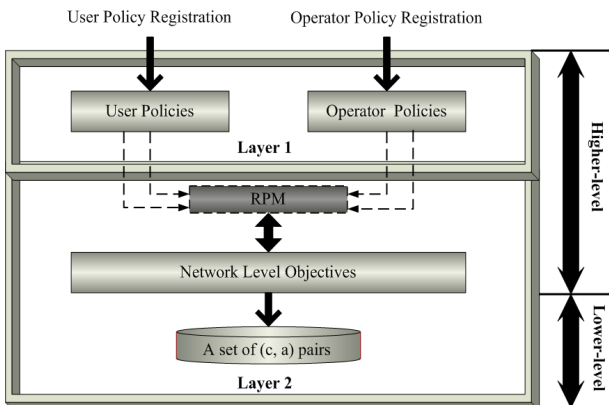


Figure 2. Proposed two-layer policy model.

network monitoring components. The proposed framework tackles this issue via posing it as a problem of learning from the current system behavior and using the results of this learning process to dispatch new policies at runtime. Nevertheless, the main challenge in dispatching policies lies in the decision on the appropriate policy actions that can be applied to the different environments. More precisely, the following three processes arise in the decision making process, which are *policy registration*, *policy dispatch* and *policy reassessment*. This section provides the detailed information about the three processes. Figure 3 shows the relationship of the three processes.

5.1 Policy Registration

The policy registration process handles the registration request messages coming from the environment and the other two processes, i.e. policy dispatch process and reassessment process (see Figure 3). If the request messages come from the environment, the process will detect/resolve the policy conflicts and refine the abstract higher-level objectives to lower-level policies. If the requests come from the other two processes, it generally means that the lower-level policies should be constructed. The registration procedure firstly reads higher-level objectives then constructs lower-level policies and finally adds them to policy repository. Once receiving the request messages, policies are constructed based on the current conditions. Note that before storing the new constructed policies, policy conflict detection and resolution are performed.

5.2 Policy Dispatch

The Policy dispatch process is responsible for deciding policies to be dispatched, which is triggered by the environment. By monitoring the heterogeneous systems context information, the network/terminal monitor triggers the reconfiguration policy dispatch process if necessary.

Once receiving a trigger messages, a judgment of whether there exist any appropriate policies is performed. Based on the result, different actions are adopted. If there are not any appropriate policies can be applied to the current context, the dispatch process sends out a policy-registration message to request to construct policy at first. After receiving a response message, the dispatch process negotiates with the neighboring RPM through *negotiation request* and *response* messages to ensure the dispatch of the most appropriate policies. Otherwise, the dispatch process directly begins the negotiation as described above. After the negotiation, the selected policies are dispatched to the environment.

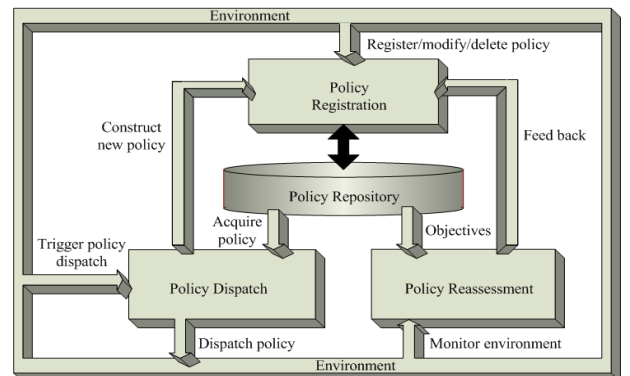


Figure 3. Proposed policy adaptation approach.

5.3 Policy Autonomic Reassessment

The policy reassessment process reassesses the existing lower policies and triggers the update if necessary. This process evaluates the previous dispatched policies periodically. In case of failing to achieve the required objectives, the result is fed back to the policy registration component, and then the policy registration process modifies the previous applied policies and constructs new policies based on the feedback. Otherwise, it continues its reassessment process.

In order to reach the optimum decision, the policy reassessment cooperating closely with policy dispatch procedure integrates RL mechanism [10-13] to be able to learn the optimal policy from its online operations. The benefit from RL is its ability to support decision strategies for unknown environments with large dimensions that constantly change. Through the “trial-and-error” interaction with the environment, the policy-based reconfiguration management approach learns to dispatch the most appropriate policy action that can be applied to the different network components.

After presenting the detailed information of the three processes, we illustrate the implementation of the policy adaptation in methodology as follows.

Firstly, a high-level business objective is refined to low-level policies, which comprise a set of condition-action pairs. Denote $C=\{c_1, c_2, \dots, c_n\}$ by the condition space of all possible trigger messages of the environment, and $A=\{a_1, a_2, \dots, a_m\}$ by the action space of all possible actions. Then the results of the refinement are a set of (c_i, a_j) value pairs, which are consequently stored in the policy repository.

The final goal of the policy adaptation process is to find and optimize the action for each trigger condition and this process is carried out by the Q-learning mechanism [14], which is a popular RL mechanism. Initially, the Q value is associated with each pair of policy rule (c, a) . In each round of iteration, the RPM perceives the triggering condition $c \in C$ of the environment and decides the action $a \in A$ following its current policy. Consequently, through policy reassessment with simple Q value iterations, the environment feeds back a reinforcement signal $r(c, a)$, called the immediate reward, to RPM. Then the policy repository generates or updates its policy according to $r(c, a)$, and enters the next round of iteration.

The adaptation process is detailed as following two steps. Firstly, the action is selected though the policy dispatch process based on Boltzmann method with p being diminished over time due to the trade-off between exploration and exploitation [10]. Secondly, $r(c, a)$ is periodically calculated and the policy pair (c, a) is modified and stored in the repository. With the autonomic policy reassessment, the Q value of each (c, a) pair converges to the maximum value, and then the optimal policy is achieved. Note that the policy iteration is independent of the Q value iteration and can be chosen flexibly if only the environment is explored sufficiently.

Policies are generated or modified by updating of the Q value, using the following iteration rule:

$$Q_{t+1}(c, a) = (1 - \alpha)Q_t(c, a) + \alpha(r_t + \gamma \max_{a'} Q_t(c', a')), \quad (1)$$

where $\alpha \in [0, 1]$ is the learning rate, $\gamma \in (0, 1)$ is the discount factor reflecting the significance of the future feedback relative to the current one. As $t \rightarrow \infty$, if the learning rate is decreased suitably to 0 and the Q value of each (c, a) pair is visited infinitely often, $Q_t(c, a)$ converges to $Q^*(c, a)$ with probability 1 [14].

6. ILLUSTRATIVE EXAMPLES

To illustrate the proposed policy reconfiguration adaptation approach discussed above, we consider a joint session admission control (JOSAC) scenario for the sessions arriving at a service area covered by multiple RATs of a single operator. Through the process of the proposed adaptation approach, the policies are generated and updated dynamically to allocate an appropriate RAT for each session. Those RATs are heterogeneous in terms of coverage and the suitability for different types of services. Assuming the terminals can connect to any RAT in the co-covered area by the means of reconfiguration [2], the high-level objective is to decide the appropriate RAT for each session to provide optimal traffic distribution according to the service requirement and the resource availability in the multi-radio environment.

The key issue arise in the policy-based management is related to the basic steps taken to reach the optimum decision. This issue is addressed and inspired by approaches to the process of policy making. In these approaches, the policymaking process passes through three main phases: policy refinement, candidate actions selection, and finally, reassessment of the applied decision. As shown in Figure 4, the proposed examples follow the same steps to decide network policies, while a feedback mechanism is used to ensure the correctness of the delivered policies. The following sections present details of these three phases and their implementations.

6.1 Policy Refinement

The first step in the adaptive policy making process is policy refinement. At this stage, the higher-level abstract objectives are refined to lower-level operational policies, i.e. a set of condition-action pairs. We define the condition space C and action space A as follows.

Condition Space: A series of discrete-time session arrivals and departures constitute the major events that affect the conditions of the dynamic environment. Since the departures will not trigger any RPM control actions in our problem, we associate the states with only the session arrivals for simplicity as in [10] and give the definition as:

$$C = (y, n, h, v, l), \quad (2)$$

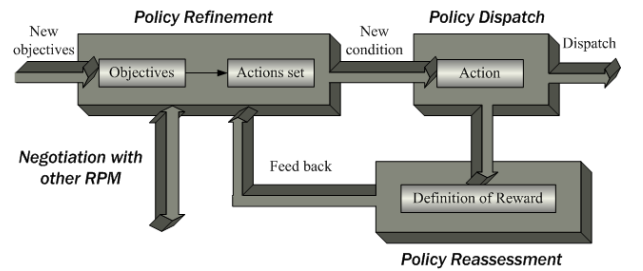


Figure 4. Details of the proposed adaptation example.

where $y \in \{0,1\}$ denotes whether a request session should be redirected to other overlapped capable RATs. Note that a session is able to be redirected at most one time in order to reduce the signaling overhead and access delay. $n \in \{1,2,\dots,K\}$ denotes the number of visible RATs (i.e. the coverage condition of the terminal that is initiating the session), where K is the total number of overlapping RATs. $h \in \{0,1\}$ distinguishes a new arrival from a handover request, $v \in \{1,2,\dots,N\}$ denotes the requested type of service, $l \in \{0,1\}$ is the current load distribution, which is generally continuous. This makes it unrealistic to represent the Q values in a lookup table so that certain generalization technique [10] must be taken to compact the learned information. The detailed generalization process is not concern of this paper.

Action Space: The action to be taken upon each session arrival is to select a RAT to admit the request or just reject it as:

$$A = \{0,1,2,\dots,K\}, \quad (3)$$

where the action of 0 denotes rejection while other non-zero actions $a = k \in \{0,1,2,\dots,k\}$ denotes that the session is admitted (or redirect) to the RAT k .

When a new session request arrives at the co-covered service area, the RPM should collect the load information of its managed RATs and the feature of the arriving session to construct current condition c following (2). Thus we need to set the discount factor γ and the initial learning rate α_0 for the iteration, and the initial exploration probability p_0 for action selection.

6.2 Policy Dispatch

The policy iteration is independent of the Q value iteration and can be chosen flexibly if only the environment is explored sufficiently. Using the Boltzmann method, an output action a is selected from $A = \{0,1,2,\dots,K\}$ in condition c with the following probability:

$$p(a|c) = \frac{e^{Q(c,a)/T}}{\sum_{b \in A} e^{Q(c,b)/T}}, \quad (4)$$

where T is *Temperature* parameter. As the iteration going on, $T \rightarrow 0$. Then selected action is executed by the RPM according to the meaning defined in (3) and is recorded for the Q value updating later.

6.3 Policy Reassessment

The final step in the policy adaptation process is performed in order to reassess the previously dispatched policy.

The policy reassessment behavior provides the opportunity to improve the online performance through a feedback from the environment. We define the feedback (e.g. the *immediate reward* in RL) by taking action a in condition c :

$$r(c,a) = \eta(v,k)\beta(h)\Delta_i, \quad (5)$$

where Δ_i is the service time for each session, $\eta(v,k)$ is the service revenue coefficient (also called matching coefficient) embodies the suitability of RAT k to the traffic type v of each

session. $\beta(h)$ is the handover reward gain, which gives more reward to the sessions which admit handovers, so as to reduce the handover dropping probability. $\Delta_i = 0$ denotes that the request session is rejected.

By pursuing the maximization of the long term reward based on the immediate reward defined in (5), the RPM will finally be wise to avoid the action of rejection that has zero reward and to select the action that is the most profitable.

In addition, in order to improve the cooperation among the different RATs, a reward sharing parameter δ is introduced to allocate appropriate resources for different types of sessions. If the redirected session is admitted by the target-RAT, the reward in (5) will be shared by the original-RAT and the target-RAT with the possibility $(1-\delta)$ and δ . The total system revenue is a summation of cumulative reward of each RAT.

The details of policy reassessment process are explained as follows: Before and after the selected action being executed, the immediate reward is obtained as the difference according to (5). Once the next trigger condition c' becomes available as a new session arrives, the new set of Q values $Q(c')$ can be produced. Having the recorded action and the resulted reward, the buffered $Q(c)$ is updated as $Q_{t+1}(c)$ by (1). At the end of each round of iteration, the learning rate α and the Temperature T need to be updated. As mentioned before, both of them are required to be decreased gradually to zero as the learning process goes on. We set them diminished exponentially with respect to the number of iterations.

Such iterations can continue until α and T reach zero or be terminated when certain pre-defined threshold (e.g. the number of iterations) is reached. By then, the policies for joint admission control are autonomously learned by the RPM. Since the implementation doesn't require any prior knowledge about the traffic model, it's adaptive to different environments.

7. SIMULATION DETAILS AND RESULTS

7.1 Simulation Configurations

This section simulates the illustrative examples above, evaluating the performance of the proposed framework. A simulation scenario was constructed to model a multi-radio scenario, which consists of three overlapped cells: GSM/GPRS, UMTS and WLAN, as shown in Figure 5. The performances of blocking probability, dropping probability and network profit are investigated. The related simulation parameters are listed in Table 1.

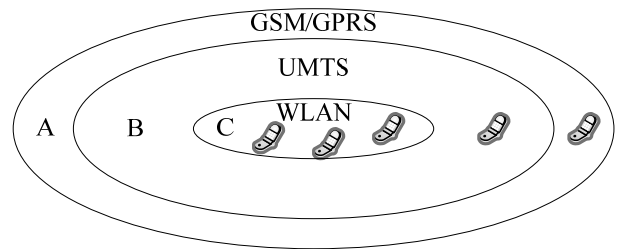


Figure 5. Simulation scenario.

Table 1. Simulation configuration

		GSM/GPRS	UMTS	WLAN
Cell capacity (kbps)		200	800	2,000
$\eta(v,k)$	Voice	3	5	1
	Data	3	1	5
		Area A	Area B	Area C
Arrival Distribution		0.1λ	0.1λ	0.8λ
		New Call		Handover
$\beta(h)$		1		10
$\delta=0.2$		$\gamma=0.5$	$T_0=30$	$\alpha_0=0.5$
$N=2$		$K=3$	Iteration times: 20,000	

Two types of services, i.e. voice and data, are assumed in the given service area. Both of them require the same bandwidth of 32 kbps and has equal arrival pattern. The total session arrivals follow the *Poisson* distribution with the mean arrival rate of λ calls/hour. The session duration is exponentially distributed with the mean value of $1/\mu=120$ seconds. Empirically, we suppose the central area, i.e. Area C, is the hot spot which has a relatively high arrival rate (0.8λ), while Area A and B share the rest arrivals uniformly. Besides, 15% arrival sessions are simulated as

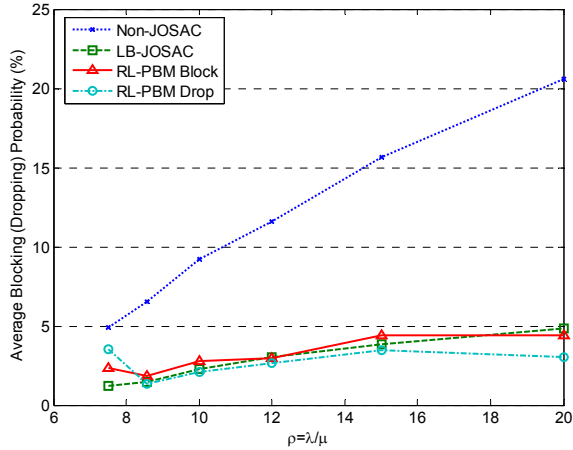


Figure 6. Blocking and dropping probability performance.

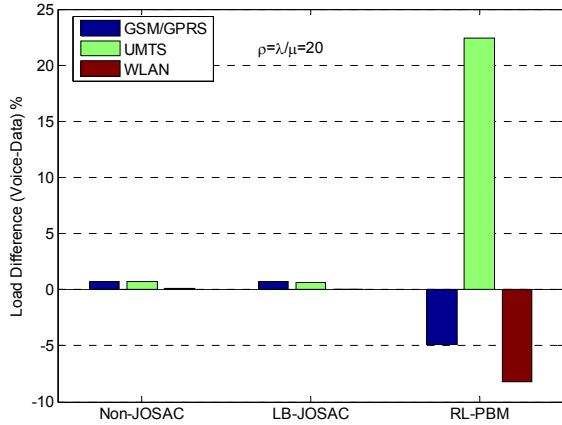


Figure 8. Load difference (voice and data) in different RATs ($\lambda/\mu=20$).

handover requests in the multi-radio scenario. The load distribution in (2) is unified quantified to $l \in \{1, 2, \dots, 10\}$.

Two reference schemes are simulated for comparison, i.e. Non-JOSAC and LB-JOSAC. When reaching the network capability, Non-JOSAC will reject all the new-arrival sessions, while LB-JOSAC direct the new arrivals to the least loaded RAT for load balancing purpose [20]. Note that the LB-JOSAC scheme always has the lowest blocking probability (approximately zero), since it balances the load among RATs so that none of the RAT is easy to be saturated.

7.2 Results

Firstly, we show the blocking and the dropping probability in Figure 6. Since Non-JOSAC and LB-JOSAC do not distinguish the new arrival from the handover request, the blocking and the dropping probability are almost the same. Comparing to Non-JOSAC, RL-PBM can reduce the blocking probability to a certain extent like LB-JOSAC, whereas it gets much lower handover dropping probability than LB-JOSAC. The reason is that RL-PBM can reject the new arrival sessions and allocate resources to the following handover sessions, i.e. achieves the better dropping probability performance while suffers the worse blocking probability performance.

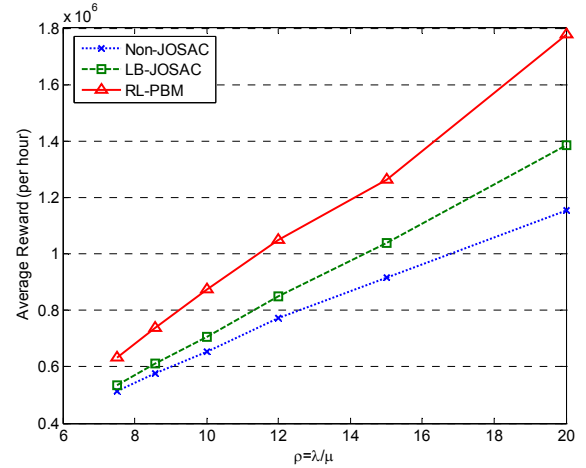


Figure 7. Average reward profit (per hour).

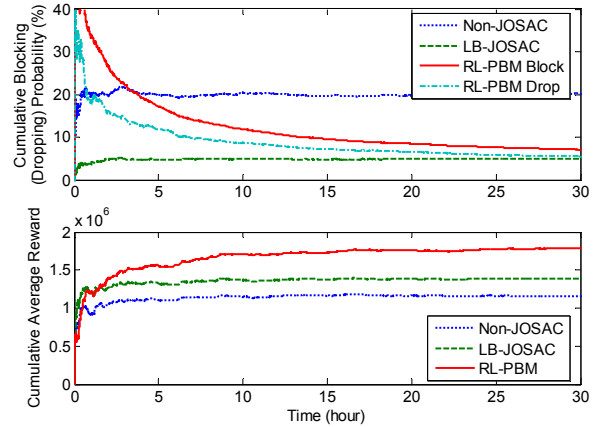


Figure 9. Online performances ($\lambda/\mu=20$).

Next, we discuss the average reward profit of the three schemes as in Figure 7. As expected, the performance of RL-PBM is superior to the other two schemes. The main reason is that through online learning, RL-PBM can redirect different types of services to their most appropriate RATs, i.e. the RATs which get the most rewards, so as to reach the optimum use of the system resource. This can be shown more specifically in Figure 8. As seen in Figure 8, the voice and data service distribution are almost the same in Non-JOSAC and JB-JOSAC, whereas in RL-PBM, the resources are allocated adjust to the attributes of the different RATs: The proportion of the voice service is much higher than that of the data service in UMTS; on the contrary, data service holds the majority position in WLAN. Consequently, the best RAT-service matching contributes to the most efficient and sufficient resource utilization so as to improve the whole system profit.

Finally, we examine the online behavior of the proposed policy-based framework by showing its cumulative performances. The online learning performances in terms of both the cumulative blocking probability and the cumulative average reward are shown in Figure 9 comparing to the first several hours, in which the RPM is “blind” to find the optimum policy, the experience learned is exploited for more rational actions during the last few hours.

8. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel framework for adaptive policy-based reconfiguration management in heterogeneous networks. The novelty of the presented work lies in the policy adaptation approach to reduce the human intervention by applying the RL methodology. It has been shown that adapting policies at runtime provides the opportunity to improve the online performance by exploiting the past experience. In addition, the policy hierarchy has been proposed to refine the higher-level requirements into lower-level policies to govern network elements’ reconfiguration, which gives more freedom to users and operators to describe their requirements in a continuously changing manner. Simulation results demonstrated the effectiveness of the proposed work. In the future, we plan to further evaluate the performance of the proposed framework through a prototype implementation.

9. REFERENCES

- [1] M. Dillinger, K. Madami, and N. Alonistioti (Editors), *Software Defined Radio: Architectures, Systems and Functions*, John Wiley & Sons Ltd, 2003.
- [2] EU IST Project E²R II (End-to-End Reconfigurability – Phase 2), <http://www.e2r.motlabs.com/>
- [3] R. Chadha, et al, “Policy-Based Mobile Ad Hoc Network Management”, in *Proceedings of the Fifth IEEE Workshop on Policies for Distributed Systems and Networks (POLICY’04)*, New York, USA, June 2004.
- [4] B. Moore, E. Ellesson, J. Strassner, A. Westerinen, “Policy Core Information Model Version 1 Specification”, RFC3060, February 2001.
- [5] N. Samaan, A. Karmouch, “An Automated Policy-based Management Framework for Differentiated Communication Systems”, *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 2236 – 2247, Dec. 2005.
- [6] You-Sun Hwang, Eung-bae Kim, “An architecture of SNMP-based network management of the broadband wireless access system”, in *Proceedings of the 9th Asia-Pacific Conference*, vol.3, pp.1163-1166, Sept.2003
- [7] K.Yoshihara,M.Isomura, and H.Horiuchi, “Distributed Policy-Based Management Enabling Policy Adaptation”, *IEICE Trans. on Communications*, vol. E87-B No.7, pp. 1854-1865, Jul. 2004.
- [8] H.Chaouchi, G.Pujolle, “A New Handover Control in the Current and Future Wireless Networks”, *IEICE Trans. on Communications*, vol.E87-B No.9, pp.2537-2547, Sep. 2004.
- [9] A.K. Bandara, E.C. Lupu, J. Moffett, A. Russo, “A Goal-based Approach to Policy Refinement”, *Proceedings of the Fifth IEEE Workshop on Policies for Distributed Systems and Networks (POLICY’04)*, New York, USA, June 2004.
- [10] L. P. Kaelbling, M. L. Littman, X. Wang et al., “Reinforcement Learning: A Survey”, *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, May 1996.
- [11] J. Nie, S. Haykin, “A Q-learning-based Dynamic Channel Assignment Technique for Mobile Communication Systems”, *IEEE Trans. on Vehicular Technology*, vol. 48, pp. 1676-1687, Sept. 1999.
- [12] S.-M. Senouci, G. Pujolle, “Dynamic Channel Assignment in Cellular Networks: A Reinforcement Learning Solution”, *10th Int. Conf. on Telecomm.*, vol. 1, pp. 302-309, Mar. 2003.
- [13] N. Lilith, K. Dogancay, “Dynamic Channel Allocation for Mobile Cellular Traffic Using Reduced-state Reinforcement Learning”, *IEEE WCNC 2004*, vol. 4, pp. 2195-2200, Mar. 2004.
- [14] C. J. C. H. Watkins, P. Dayan, “Q-learning”, *Machine Learning*, vol. 8, pp. 279-292, 1992.
- [15] Distributed Management Task Force, “CIM Policy Model, v.2.12,” http://www.dmtf.org/standards/cim/cim_schema_v212/cimv212Final-Doc.zip, Apr. 20, 2006.
- [16] J. Vogler, et al., “Equipment Management and Control Architecture”, July 2005, E2R White paper, available at <http://e2r.motlabs.com/whitepapers/>
- [17] J. Shin, J. W. Kim, and C. J. Kuo, “Quality-of-service mapping mechanism for packet video in differentiated services network,” *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 217–230, Jun. 2001.
- [18] G. Ghinea, J. P. Thomas, and R. S. Fish, “Mapping quality of perception to quality of service: The case for a dynamically reconfigurable communication system,” *J. Intell. Syst.*, vol. 10, no. 5/6, pp. 607–632, 2000.
- [19] J. Shin, J. W. Kim, and C. C. Kuo, “Quality-of-service mapping mechanism for packet video in differentiated services network,” *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 219–231, Jun. 2001.
- [20] M. Alanyali, B. Hajek, “On Simple Algorithm for Dynamic Load Balancing”, *IEEE INFOCOM’95*, vol. 1, pp. 230-238, Apr. 1995