Multi-Agent Reinforcement Learning in Network Management

Ricardo Bagnasco and Joan Serrat

Network Management Group, Departament de Teoria del Senyal i Comunicacions.
Universitat Politècnica de Catalunya.
{rbagnasco, serrat}@tsc.upc.edu

Abstract. This paper outlines research in progress intended to contribute to the autonomous management of networks, allowing policies to be dynamically adjusted and aligned to application directives according to the available resources. Many existing management approaches require static *a priori* policy deployment but our proposal goes one step further modifying initially deployed policies by learning from the system behaviour. We use a hierarchical policy model to show the connection of high level goals with network level configurations. We also intend to solve two important and mostly forgotten issues: the system has multiple goals some of them contradictory and we will show how to overcome it; and, some current works optimize one network element but being unaware of other participants; instead, our proposed scheme takes into account various social behaviours, such as cooperation and competition among different elements.

Keywords: machine learning, policy-based management, adaptive policies, autonomic communication system.

1 Introduction

The growth of Internet and particularly the rapid advances in real time supported applications that are expected to be developed, make its management a major challenge. Among the different enabling technologies for autonomic communications, the policy based management is one of the most representatives. This paradigm allows the segregating of the rules that govern the behaviour of the managed system from the functionality provided by the system itself [1]. The most developed and contemporary implementations of the paradigm rely on pre-programmed rules based on logic. Nevertheless, to be called autonomic, a system must show a degree of flexibility to self adapt to changes in the goals, services or resources and in our opinion this is hardly achievable by means of static policies. In contrast, we propose the control of the system behaviour by means of dynamic policies; that is, policies that are allowed to change according to the evolution of the system. In short, we pretend to design a method that dynamically and continuously seeks for management policies that better fit the context of execution looking for its optimization. This is innovative because current approaches lack of learning capabilities and most of the recent work just mention the need of it [2], [3].

With this conception of the problem, the first challenge is to figure out the learning mechanism or mechanisms that will lead to new policies. The second, but no less important is how to solve the potential conflicts that can arise from such a policy generation process. Finally, it would be important to understand the whole vertical structure from devices till applications in order to combine and enforce the best configuration. All these challenges will be addressed in this work in the following manner: for the first two issues we plan to use a model free learning algorithm and because of the size of the network we will investigate multi-agent variations, especially those that acknowledge the existence of other competitive/cooperative agents in the same environment. For understanding the whole structure of policies we will define and use a policy architecture from the *Application level* to the *Device level* in a similar approach as the Policy Continuum [4] showing the relationships between policies of different levels.

The remainder of the paper is structured as follows: section 2 provides an overview of work related to our topic of interest. In section 3 we introduce our idea and show our planned topics of research. Section 4 finishes this article with the overall conclusions.

2 Related work

Since managing a complete network involves several topics, we could find related work in a wide area of research. We will focus in communication systems but, for instance in the multi-hardware configuration [5] or sensor networks [6] we can find analogous problems too. And because we will use learning techniques we could also find related work (in an abstract way) in the machine learning field [2].

There are a few examples of approaches for a complete solution of learning in network management. A work that could be considered as a starting reference is described in [7]. It is essentially focussed in policy refinement but, from the point of view of what we pretend to do, it doesn’t consider dynamic adaptation of policies in the presence of dynamic environments. Another approach to introduce an architecture oriented to an adaptable service is in [8]. Here the authors use rule-based reasoning and extended finite state machines. Their mechanism to select the rule is using a *Reasoning Procedure*. One disadvantage of their solution is the need of a careful specification of all possible events. A different approach can be found in [9], where they propose some similar goals as in our work (adaptation and flexibility) and also use a policy hierarchy. But they skip the multi-goal issue and they just sketch some framework but don’t explain the details of their mechanisms such as conflict resolution because of the coexistence of other agents. In contrast, although we also pretend to follow an adaptation process based on learning, our target is a more flexible way of adaptation of policies to changes in goals and the environment with embedded policy conflict avoidance.

3 Planned Approach

We consider a continuum of policies [4] constituted by several layers; in the simplest form by only two levels. The lowest level corresponds to *device policies* managing the physical or virtual resources. At this level we have some configurations that the device can offer to the applications so the higher level can choose between those pre-configurations. The second level contains *application policies*. We understand that applications use the resources of device level. For example we could have a printer with three configurations: *Draft*, *Black_and_White*, *Full_Color*; and different applications could prefer one setting over another. Or for instance some links could have two metrics: *speed of transmission* and *error rate*, and offer five pre-configurations with different values of speed and error rate, having a different impact in the performance of the applications so it is needed to find the best option in general.

Because the algorithm we plan to use could be quite slow to converge it would be practical to initialize the policies at device level with some initial state (for instance with the output of some simulation of the situation). After this initialization the policies will be evolved online (in the real world) by means of Reinforcement Learning [10], a sub area of Machine Learning, concerned with how an agent should take actions in an environment to maximize some long-term reward. In particular we are interested in the Temporal Difference techniques and Q-Learning [11] more specifically. This technique learns an action-value function to estimate the expected utility of taking some action in a given state. Applications should give some feedback (reward) to its devices in order to inform about how good or bad its performance is to their own goals. Taking this information into account the lower level will know which action (pre-configuration over its metrics) is the best for the system at each moment. The changes in environment, devices, and applications will affect the performance making our previous preferred pre-configuration probably no longer optimal so our algorithm will notice that and adjust it to a new selection that is the best for that moment and situation. The ultimate goal will be to maximize the reward of the system, for example evaluating a weighted sum of the application's rewards, and it will be very important to pay special attention to the carefully design of that function. In addition, because it is not feasible to consider the whole system as a single agent, we plan to use multi agent systems and study several distributed algorithms in order to better solve our problem. We plan to validate our proposal using the AutoI infrastructure [12] in one of its general use cases and, eventually, using the OPNET simulator [13] extending it to use a policy-based management system. But first we should make a proof of concept showing that changes in environment cause changes in politics and then we should measure time of convergence to stable politics in several kinds of environments evaluating different algorithms.

4 Concluding remarks

This paper has described our ongoing research work towards managing a communications system in a flexible and adaptable way and configuring the devices aligned to user and application goals. We tackle most of the open issues identified in the most recent papers and surveys [2]. Specifically the following challenges are considered: from supervised to autonomous learning; from offline to online learning; from fixed to changing environments and from centralized to distributed learning. Our immediate future work will be the definition of a scenario in which we demonstrate the feasibility of our approach and the investigation of several algorithms to choose the best suited one.

We strongly believe that, because of network complexity, learning policies is key in network management in contrast to knowledge based approaches. With this regard, our contribution is to the best of our knowledge, the first proposal to learn, adapt and align policies from the devices configuration till the application taking into account the presence of other entities competing for resources and having their own goals.

Acknowledgments. This research is supported in part by the "Programa de FPI del Ministerio de Innovación y Ciencia de España". This is also part of the AutoI project and of the EMANICS NoE.

5 References

1. R.Boutaba and J. Xiao. "Network Management: State of the Art". *Communication Systems: The State of The Art (IFIP World Computer Congress), pp127-146.* 2002
2. T. G. Dietterich and P. Langley. "Machine Learning for Cognitive Networks: Technology Assessment and Research Challenges*". Chapter 5 of the book: Cognitive Networks: Towards Self-Aware Networks. 2007*
3. N. Badr, A. Taleb-Bendiab and D. Reilly. "Policy-Based Autonomic Control Service". *Fifth IEEE International Workshop on Policies for Distributed Systems and Networks 2004*
4. S. Davy, B. Jennings and J. Strassner. "The policy continuum-Policy authoring and conflict analysis". *Computer Communications. Volume 31, Issue 13, pages 2981-2995* August 2008.
5. J. Wildstrom, P.Stone, E. Witchel, R.J. Mooney and M. Dahlin, "Towards Self-Configuring Hardware for Distributed Computer Systems*". Proceeding of the Second International Conference on Autonomic Computing. ICAC'05*
6. I.F. Akyildiz, S.Weilian, Y. Sankarasubramaniam and E. Cayirci. "A survey on sensor networks". *IEEE Communications Magazine. Vol. 40 Issue 8. Pages 102-114*. Aug. 2002.
7. A.K. Bandara, E.C. Lupu, J. Moffett and A. Russo. "A Goal-based Approach to Policy Refinement". *Proceedings 5th IEEE Workshop on Policies for Distributed Systems and Networks*. June 2004
8. P. Supadulchai and F. Arve Aagesen, "Policy-based Adaptable Service Systems Architecture". *21st International Conference on Advanced networking and Applications . IEEE* (AINA'07).
9. N. Samaan and A. Karmouch. "An Automated Policy-Based Management Framework for Differentiated Communication Systems". *IEEE Journal on Selected areas in communications, Vol. 23, Nº 12*, December 2005
10. R. Sutton and A.S. Barto. "Reinforcement Learning: An introduction". *MIT Press, Cambridge, MA*. 1998
11. C.J.C.H. Watkins and P. Dayan. "Technical Note: Q-Learning", *Machine Learning 8: 279-292*. 1992
12. Autonomic Internet Project: www.ist-autoi.eu/autoi (2009)
13. www.opnet.com (2009)