A Fuzzy Reinforcement Learning Approach for Pre-Congestion Notification Based Admission Control

Stylianos Georgoulas[1], Klaus Moessner[1], Alexis Mansour[1], Menelaos Pissarides[1], and Panagiotis Spapis[2]

[1] Centre for Communication Systems Research,
Faculty of Engineering and Physical Sciences,
University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom
{s.georgoulas,k.moessner,a.mansour,m.pissarides}@surrey.ac.uk
[2] Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens,
Panepistimiopolis Ilissia, Athens, 15784, Greece
pspapis@di.uoa.gr

Abstract. Admission control aims to compensate for the inability of slow-changing network configurations to react rapidly enough to load fluctuations. Even though many admission control approaches exist, most of them suffer from the fact that they are based on some very rigid assumptions about the per-flow and aggregate underlying traffic models, requiring manual reconfiguration of their parameters in a "trial and error" fashion when these original assumptions stop being valid. In this paper we present a fuzzy reinforcement learning admission control approach based on the increasingly popular Pre-Congestion Notification framework that requires no *a priori* knowledge about traffic flow characteristics, traffic models and flow dynamics. By means of simulations we show that the scheme can perform well under a variety of traffic and load conditions and adapt its behavior accordingly without requiring any overly complicated operations and with no need for manual and frequent reconfigurations.

Keywords: Admission Control, Pre-Congestion Notification, Fuzzy Logic, Reinforcement Learning, Quality of Service, Autonomic Management.

1 Introduction

The envisioned dynamicity of future Internet networks, where applications with different service requirements may appear makes Quality of Service (QoS) provisioning and service continuity a challenging issue that traditional traffic engineering approaches, usually based on offline optimizations through bandwidth provisioning, may not be able to address efficiently. Towards this end, dynamic service management functions such as admission control can play a significant role with respect to supporting QoS for application flows during the actual service delivery time, helping to overcome the inability of slow-changing network configurations to react adequately fast to shorter-term load fluctuations.

R. Sadre et al. (Eds.): AIMS 2012, LNCS 7279, pp. 26–37, 2012.

Even though admission control is a well-studied subject [1, 2], most of the existing schemes suffer from the fact that they are based on some very rigid assumptions about the per-flow and aggregate underlying traffic models, requiring therefore manual reconfiguration of their parameters in a "trial and error" fashion as soon as the original assumptions stop being valid, in order to keep performing well [3]. That is they employ some tuning parameters that need to be initially manually set and also readjusted as soon as the traffic and network characteristics change.

The idea of mechanisms able to self-adapt and self-configure as the conditions change has been around for quite some time under the generic term *autonomic management* and has been gaining steadily increasing interest during the past few years. In that context, past and existing projects [4] have been working towards inducing self-* behavior in Internet communication mechanisms. In this direction, in this paper we propose a novel, autonomic admission control scheme based on the increasingly popular Pre-Congestion Notification (PCN) framework put forward by IETF [5]. The proposed solution adapts autonomically to the characteristics of the traffic flows and underlying network traffic and can perform well under a variety of traffic and load conditions without making any assumptions about traffic models, flow dynamics and characteristics and with no need for manual and frequent reconfigurations.

The rest of this paper is organized as follows; in Section 2 we present the underlying concepts behind PCN, the variations of this approach and its limitations, as well as some existing approaches towards introducing autonomic behavior into PCN. In Section 3 we present in detail our scheme and in Section 4 we evaluate its performance under a variety of traffic and load conditions. Finally, in Section 5 we conclude, summarizing our findings, while we also give some directions for future work.

2 Pre-Congestion Notification Based Admission Control

PCN, which targets core/fixed network segments, defines a new traffic class that receives preferred treatment by PCN-enabled nodes, similar to the expedited forwarding per-hop behavior in Differentiated Services [6], aiming to minimize the packet loss rate (PLR) for loss-intolerant flows. The PCN framework provides two main functionalities that are admission control (AC) and flow termination (FT) [7]. AC, as also aforementioned, decides on whether new flow requests should be admitted or rejected based on the current network conditions whereas FT is a control function that tears down already admitted flows in case of overloads that might occur, in spite of AC, due to rerouted traffic (i.e. in case of link failures and other unexpected events). AC targets the "normal operations" phase of a network whereas FT can be seen as a radical measure for use only in emergency situations and should be avoided as much as possible (it is more acceptable to deny a flow session than to allow it to start with high uncertainty about the chances of completing, which renders the session useless [8]).

In order to support both these functionalities PCN introduces an admissible and a supportable rate threshold (*AR(l)*, *SR(l)*) ***for each link*** *l* of the network, which create three different load regimes. If the PCN rate *r(l)* is below *AR(l)*, there is no pre-congestion and -from that link's point of view- further flows can be admitted in the ingress-egress path(s) to which it belongs to. If the PCN traffic rate *r(l)* is above *AR(l)*, the link is AR-pre-congested and no further flows should be admitted

depending also on how much the rate exceeds *AR(l)*. If the PCN rate *r(l)* is above *SR(l)*, the link is SR-pre-congested and in this state some of the existing flows should be additionally terminated, depending also on how much the rate exceeds *SR(l)*. Both the AC and FT mechanisms are triggered based on packet markings; that is PCN nodes mark traffic accordingly depending on whether it exceeds *AR(l)* or *SR(l)*, the egress nodes evaluate the packet markings and deduce on the admission control and -if needed- flow termination decisions. The above are illustrated in Fig. 1.

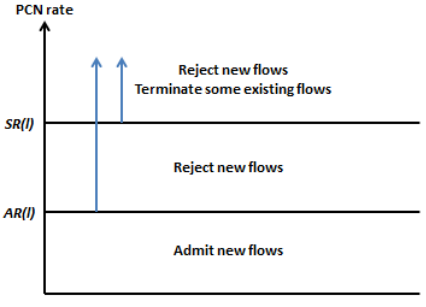

Fig. 1. PCN rates and behavior

PCN-based AC can be performed in various ways [7]. In its Probe Based AC (PBAC) version the markings on probe packets only are evaluated and the admission control decision is derived. In the Observation Based AC (OBAC) version, no probe packets are sent and a single marked packet of the "main flows" aggregate traffic is considered enough to set the AC status for the involved ingress-egress pair(s) to *reject* for the subsequent time period. Finally, in the Congestion Level Estimate (CLE) Based AC (CLEBAC) version, no probe packets are sent and at regular intervals the percentage of marked versus total packets of the "main flows" aggregate traffic is evaluated and the AC status is set to *accept* or *reject*, depending also on whether the CLE value is below or above a predefined threshold value.

The main deficiency of PCN-based AC is that even though the possible marking behaviors and the possible AC mechanisms are described in detail [5, 7], the way to actually set the marking thresholds so as to achieve the desired QoS targets is not addressed. While there has been considerable work [9, 10, 11] in evaluating the performance of the various versions of PCN-based AC schemes in (mostly) single link topologies, these works assume that the marking thresholds and the other involved PCN parameters (e.g. the CLE threshold value) can also be derived and set in the first place; in principle they evaluate the performance of the schemes as a function of their involved parameters without though providing any guidelines on how these parameters should be set if the schemes were to be applied in a practical networking scenario. This means that in order to apply PCN-based AC even for a single path consisting of 10 links there exist 10 distinct marking threshold values that need to be manually adjusted so that the combined marking behavior along all these links, when used in the AC mechanism, guarantees the desired QoS targets. Thus, when network characteristics such as links capacities and/or flows characteristics change, these thresholds have to be manually readjusted until these QoS targets are again met. In principle a

network administrator would have to continuously monitor and intervene and update the parameters until an acceptable level of performance is acquired. Apart from the added human effort this would also mean considerable amount of downgraded performance until the manual parameter fine-tuning led to the correct parameter settings.

These shortcomings were recently acknowledged and some approaches towards inducing autonomic behavior in PCN-based AC have emerged [12, 13, 14]. The main concept behind these CLEBAC approaches is that the maximum traffic variability can be considered as a good metric to be taken into account when setting the marking threshold. If the marking threshold at one link is set as equal to *link capacity – maximum traffic variability*, then packet marking would occur only when indeed there is no "space" left for any additional traffic without risking packet losses. To account for the fact that the future maximum traffic variability may be different than the existing measured one, the approaches further reduce the marking threshold by multiplying the existing maximum traffic variability with a Δ value [12, 13] and also by "adding" the traffic demands of future flows [14]. These concepts are shown in Fig. 2 which is borrowed from [12] (GoalRate corresponds to the link capacity).

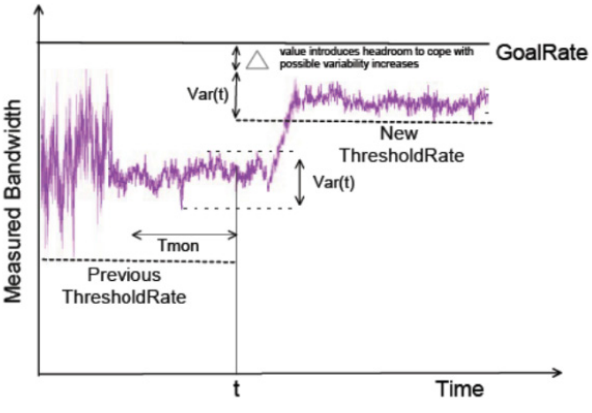

Fig. 2. Marking threshold adjustment based on maximum traffic variability [12]

Apart from the fact that the correct Δ value needs to be manually set, these approaches also ignore the effect of buffering in routers when setting the marking threshold, as well as don't account for any flows terminating (in all evaluation scenarios once a flow is admitted it is assumed that it doesn't terminate). In other words these works evaluate only how fast the schemes start reacting to congestion being built up in a constantly increasing load environment. That is they focus on the very limited time period during which the network moves from uncongested to congested conditions but they do not try to capture the long-term performance of the schemes in environments with flows arriving and terminating. In addition from a practical implementation point of view they require calculation of the maximum traffic variability and adjustment of the marking threshold on a per packet basis (in Gbps links this would mean millions of times per second) as well as keeping track and updating of the flow characteristics at all routers, even core ones [14].

In our scheme, which will be described in detail in the following Section, we aim to address the long-term performance of PCN-based AC in environments with flow arrivals and terminations without inducing significant processing overhead (calculations and adjustments many times per second) or requiring any state or flow statistics being kept and updated at any routers. Through the use of machine learning, the AC controller is able, starting from some default marking threshold value, to converge to a set of rules that drive the marking behavior and threshold value readjustment, autonomically and on a *per scenario* basis as the traffic and network characteristics require.

3 Fuzzy Q-Learning PCN-Based Admission Control

Fuzzy Q-Learning has been extensively used in the literature, e.g. see [15, 16] to introduce autonomic capabilities in network control systems, and is a combination of fuzzy logic [17] with Q-learning (type of Reinforcement Learning (RL)) [18] that aims to combine the robustness of a rule based fuzzy system with the adaptability and learning capabilities of Q-learning. In this Section we highlight the main concepts and benefits of this approach and its applicability in the context of PCN-based AC.

3.1 Fuzzy Q-Learning Concepts

Rule based fuzzy systems have been extensively applied with success in many diverse application areas due to their similarity to human perception and reasoning, their intuitiveness and their simplicity. The main concept is that, contrary to classical set theory, the sets used for representing their input and output parameters are fuzzy; meaning that their elements have degrees of membership that represent the degree of truth of a statement. The process of mapping the input values into membership functions (MFs) is called "fuzzification". After this first step, membership functions are combined in fuzzy "if…then" rules to make inferences and finally the "defuzzification" phase produces a crisp output value. In principle the idea of fuzzy inference systems is that at every point in time and for a unique set of input parameter values, multiple rules can be triggered with a different degree of truth (strength) and their individual outputs are then "combined" to derive a unique crisp output value. Fuzzy inference systems offer robustness and smooth reaction [17] however they do require the existence of an expert to define the appropriate rule-set. The main challenge is therefore to be able to generate the appropriate rule-set without the existence of a direct trainer. Reinforcement learning can be applied in this context to drive the generation of the appropriate rule-set based on the interactions with the environment.

Q-learning belongs to the Temporal Difference (TD) methods which are one of the three main types of Reinforcement Learning methods, the other two being Dynamic Programming and Monte Carlo methods [18]. TD methods combine the pros of the other two types of RL methods; that is, they don't require an accurate model of the environment (contrary to Dynamic Programming) and are suitable for step-by-step incremental computations (contrary to Monte Carlo methods). Q-learning works by learning an action-value function based on the interactions of an agent (controller) with the environment and the instantaneous reward it receives. The objective of an

agent is to find, by trying out all the possible actions when being in a given state, the action that maximizes its long term reward. The detailed mathematical foundation and formulation of Q-learning can be found in [15, 16, 18] therefore it is not repeated here, due to space limitations; the core Q-learning algorithm [18] is provided though, so as to highlight the parameters involved in it and consequently in our evaluation in the following Session.

```
Initialise Q(s,α) arbitrarily (1)
  Repeat (2)
    Initialize s (3)
    Repeat (4)
      Choose α from s using policy derived from Q (e.g.
      ε-greedy) (5)
      Take action α, observe r, s' (6)
      Q(s,α)=Q(s,α)+a*[r+γ*max_α Q(s',α')-Q(s,α)] (7)
      s←s' (8)
  until s is terminal (9)
```

In every step of the Q-learning algorithm, the agent/controller observes the environment (*line 5*) and deduces in which state s it currently resides based on the input parameter values. It then (100-ε) % of the time takes the action with the highest Q value and ε% of the time it takes another action randomly from the set of available actions (*line 5*). This is called the exploration/exploitation tradeoff which aims to ensure the agent is allowed to move into "unchartered territory" instead of solely relying on what it has learnt so far. After taking the action the agent receives an instantaneous reward r, observes to which state s′ this action led it into (*line 6*) and updates the corresponding Q value (*line 7*). Parameter a $(0<a\leq 1)$ is the learning rate which determines how much the agent values the newly acquired knowledge compared with the existing one and γ $(0\leq\gamma<1)$ is the discount factor which defines how much expected future rewards affect decisions now. Low γ means the agent pays little attention to the future whereas high γ means that potential future rewards have a major influence now; that is the agent is willing to accept some short-term loss in return for long-term gain.

Q-learning is an attractive method of learning because of the simplicity of the computational demands per step and also because of proof of convergence to a global optimum, avoiding all local optima, as long as the Markov Decision Process (MDP) requirement is met; that is the next state depends only on the current state and the taken action (it is worth noting that the MDP requirement applies to all RL methods). It can also be easily combined with fuzzy logic and provide the association between the states and the available actions, which is the same as constructing the fuzzy logic "if…then" engine; the only additional step required being the distribution of the reinforcement/reward signal among multiple simultaneously triggered "if…then" rules due to the overlapping of the fuzzy input sets. This distribution can be done proportionally to the strength with which each rule is triggered so that rules (states) triggered with high strength -and contributing therefore more to the output action- are rewarded (or penalized) more compared to rules triggered with lower strength [15, 16].

3.2 Applicability to PCN-Based Admission Control

In our CLEBAC PCN-based AC approach we assume that there is a PCN agent/controller running at every PCN node and we define two input parameters for each controller and one output parameter; the input parameters are the traffic rate (as % of the link capacity) and its "trend" over the past T seconds. To define the trend of the traffic rate we split the T second interval into two intervals (t-T, t-T/2] and (t-T/2, t], calculate the traffic rate into each one individually and calculate the trend as:

$$\text{trend}_{(t-T,t)} = \text{rate}_{(t-T/2,t]} - \text{rate}_{(t-T,t-T/2]} \quad (1)$$

As output parameter we define the adjustment X of the marking threshold compared to its current setting; that is the marking threshold after each action becomes:

$$\text{ThresholdRate}_{new} = \text{ThresholdRate}_{current} + \text{defuzzified}(X) \quad (2)$$

The individual "if…then" rules for driving the marking behavior are therefore of the kind "**if** the traffic rate is VERY HIGH (POSITIVE BIG) **and** the trend is VERY HIGH (POSITIVE BIG) **then** the threshold adjustment is NEGATIVE BIG". This rule intuitively states that if the traffic is very high and keeps increasing then the marking threshold should be reduced a lot so that packet marking should be accelerated and subsequent flows should be rejected.

The membership functions for the input and output parameters used in our evaluation are summarized in Table 1. Since each of the input parameters and the output parameter can be categorized as being Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS) and Positive Big (PB) this means that the rule-set consists of 25 "if…then" rules with each rule having 5 possible output actions. The objective of the learning process is therefore for the controller to derive the optimal output action for each one of the 25 rules.

Table 1. Input and output MFs of the controller

	Traffic Rate	Trend of Traffic Rate	Adjustment (kbps)
NB	(piece-wise linear) -∞, 0.4, 0.8	(piece-wise linear) -∞, -0.5015, -0.003	(triangular) -45, -35, -25
NS	(triangular) 0.78, 0.805, 0.83	(triangular) -0.035, -0.0075, 0.02	(triangular) -38, -20.5, -3
ZE	(triangular) 0.82, 0.845, 0.87	(triangular) 0.015, 0.215, 0.28	(triangular) -5, 0, 5
PS	(triangular) 0.86, 0.895, 0.93	(triangular) 0.025, 0.0325, 0.04	(triangular) 3, 9.5, 16
PB	(piece-wise linear) 0.92, 1, +∞	(piece-wise linear) 0.035, 1, +∞	(triangular) 14, 16, 18

One important element, as in every application of RL, is the instantaneous reward function since it fundamentally defines what the controller attempts to learn to optimize in the long-term. In our case we set the reward function as follows:

$$\text{Reward} = \begin{cases} \text{Utilization, if Packet Loss Rate}=0 \\ \text{P, if Packet Loss Rate} \neq 0 \end{cases} \quad (3)$$

This way the controller attempts, based on the available actions, to learn the policy that maximizes the long-term utilization while at the same time avoiding situations

that can lead to packet loss. Setting P to a considerably high negative value can force the controller to prioritize the packet loss rate objective -which is the primary objective of PCN- over the maximization of the utilization.

One additional aspect also mentioned above is that in order to enable the learning phase, the evolution of the controlled system between successive states should be mainly due to the control action (strictly speaking, theoretically only due to the control action in order for the MDP requirement to be fully met). Since the outcome of the packet marking must be reflected in the subsequent admission control decisions, the control periodicity for the marking threshold adjustment S must be set to a reasonably high value that will allow for flows to arrive and be accepted or rejected. In the former case the evolution of traffic will be due both to new flows starting and existing flows terminating whereas in the latter case it will be due to existing flows being terminated. If, however, S is set to a very low value then the traffic evolution will be mainly (or only) due to the inherent burstiness of existing traffic; and this traffic evolution is not affected by the control action.

4 Performance Evaluation

In order to evaluate the performance of our scheme (we will be referring to it as FQL PCN) we ran simulations with the publicly available NS-2 simulator [19] using the topology of Fig. 3, in which we assume that 3 types of users request the streaming of MPEG-4 encoded video from a streaming media server through a PCN domain and at different quality levels (low, average, high) so as to have traffic sources with different bandwidth requirements and traffic characteristics in our simulations.

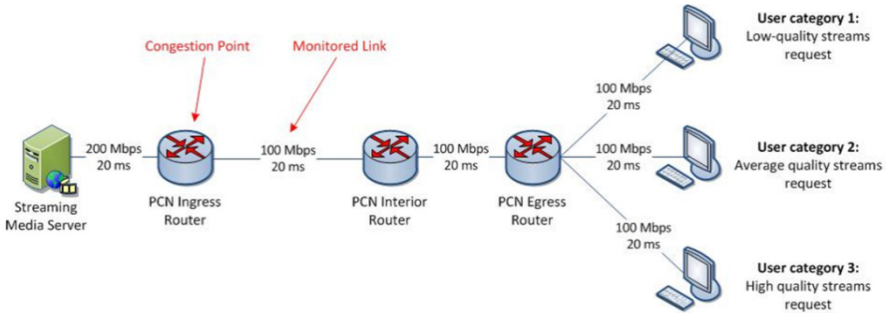

Fig. 3. Simulation topology

Since the transmission from the media server is done through a 200Mbps link and the links in the PCN domain are set to 100Mbps, congestion may occur only at the PCN ingress router, meaning that in this topology only one PCN controller is needed to monitor the state of the outgoing link and perform packet marking. In addition, given the simulation setup, the utilization of this link alone is adequate for use in the reward function, since the utilization of the downstream link (between the PCN interior router and the PCN exterior router) is identical.

Regarding the video streams we used traffic traces from [20] with the high quality streams having peak rate 3.1Mbps and average rate 0.58Mbps, the average quality streams having peak rate 1.5Mbps and average rate 0.18Mbps and the low quality streams having peak rate 1.5Mbps and average rate 0.11Mbps. To test our scheme under a variety of load conditions and flow arrival/departure dynamics we considered various scenarios with per quality type flow inter-arrival times 1sec, 2sec and 3sec (exponentially distributed) and flow durations 350sec and 1200sec, also exponentially distributed. The first duration could, for example, correspond to music video clips whereas the second to episodes of a TV series. Every scenario was run for 7200sec simulated time and 10 times with different random number generator seeds.

Regarding the scheme's parameters we fixed the monitoring interval T to be equal to 5sec, the control periodicity S to be equal to 1sec and the marking threshold was initially set equal to 95% of the link capacity. Regarding the marking behavior, we employed the exhaustive threshold marking approach [7], which marks all packets on a link when the metered rate exceeds the marking threshold rate. For the admission control decision we also assumed that whenever CLE>0 then the AC status should be set to *reject*. With respect to the Q-learning specific parameters we set P equal to -20 to "force" the controller to learn to avoid taking actions that may lead to packet losses. The discount factor γ was set equal to 0.95 and ε for the exploration/exploitation tradeoff was set equal to 0.05, meaning that with 95% probability the action with the higher Q value is taken during the learning period. Regarding the learning rate since the environment is non-deterministic (due to the probabilistic flow arrivals/departures and the inherent burstiness of the traffic, taking an action from a given state can lead to different states and rewards) we used the learning rate for the non-deterministic version of Q-learning. In this version, the practice is to begin with a higher learning rate and reduce it during the learning process by employing the following formula:

$$a(s,\alpha) = a_{max}/(1+visits(s,\alpha)) \quad (4)$$

The learning rate used to update each state-action pair ("if…then" rule) is reduced every time this state-action pair is visited; a_{max} was set equal to 0.1.

Due to space limitations only the obtained results for two scenarios will be presented; similar results and conclusions were drawn from all the tested scenarios. It is also worth noting that all the aforementioned parameters were fixed and remained unchanged throughout all scenarios and all simulation runs within each scenario.

4.1 Scenario 1 (Inter-arrival Time=1sec, Video Duration=1200sec)

Fig. 4 shows the outcomes of one out of the ten simulation runs. FQL PCN is able, after the initial period of PLR due to the high initial marking threshold, to reduce and vary the marking threshold so that PLR is equal to zero for most of the time and the utilization stays at high levels. During the initial period of PLR spikes (first 660sec on average among all simulation runs), PLR stays well below 0.5% whereas for the rest of the simulation runs the PLR spikes account in total for 12sec (on average among all runs) with a maximum PLR value of less than 10^{-4} (it is worth noting that according to [21], for MPEG-4 video this latter value of PLR can be considered tolerable).

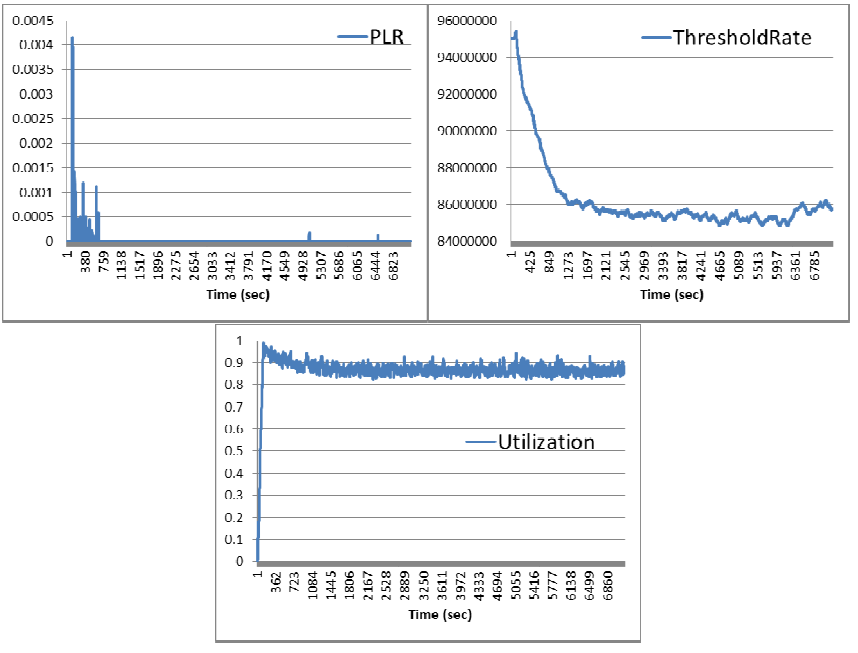

Fig. 4. Incurred PLR, marking threshold rate and utilization for the FQL PCN scheme

To illustrate the effect of learning in the rules used by the controller, in Fig. 5 we show the incurred PLR and the variation of the marking threshold rate for a fuzzy logic (FL PCN) controller that uses the same initial rules fed to the FQL controller, without though any further updates to the rules through learning.

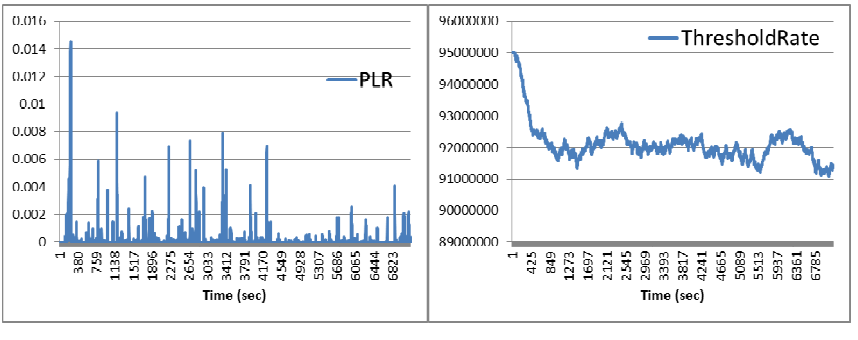

Fig. 5. Incurred PLR and marking threshold rate for the FL PCN scheme

As Fig. 5 shows for the FL PCN scheme there exist PLR spikes for the whole duration of the simulations due to the fact that the initially set rules are not optimized and are not adapted towards an optimized behavior. In contrast the FQL PCN scheme, starting by exactly the same -apparently not optimized rule-set- is able through the

learning process to redefine the rules accordingly. The FL PCN scheme achieves higher utilization (not shown due to space limitations) but since the primary PLR objective is violated, this increased utilization is of no real practical importance.

4.2 Scenario 2 (Inter-arrival Time=2sec, Video Duration=350sec)

As Fig. 6 shows for different flow arrival/departure dynamics the behavior of the FQL PCN scheme is similar to the previous scenario with an initial period of PLR spikes (reduced though due to the overall lighter load conditions) and some infrequent PLR spikes for the remainder of the simulations. The FL PCN scheme, similar to the previous scenario, is hindered by its static and not optimized rule-set.

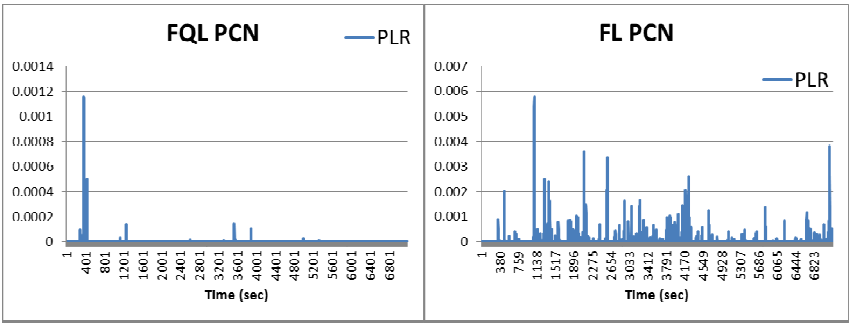

Fig. 6. Incurred PLR for the FQL PCN scheme and the FL PCN scheme

5 Conclusions and Future Work

In this paper we presented a novel scheme towards PCN-based admission control. Through the use of Fuzzy Q-learning the scheme is able to autonomically adapt its marking behavior so that it meets the traffic and network requirements. The scheme requires no *a priori* knowledge about traffic flow or network characteristics, traffic models and flow dynamics and is able to adapt accordingly and on a per scenario basis without requiring any manual reconfiguration of its involved parameters on a per scenario basis. In addition it doesn't require any complicated and overly frequent calculations or keeping and updating of any state in any routers.

In the future we will attempt to further optimize our scheme and improve its convergence properties, by examining the use of different input parameters and reward functions, assess its behavior and performance in multi-agent/controller environments required in larger, more realistic, network configurations and compare it with the existing autonomic and adaptive PCN-based AC schemes.

Acknowledgements. The research leading to these results has been performed within the UniverSelf project (www.UniverSelf-project.eu) and received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under

grant agreement no 257513. The authors would also like to thank Zwi Altman and Richard Combes from France Telecom Orange Labs and Rouzbeh Razavi from Alcatel Lucent Bell Labs Ireland for their valuable support and advice.

References

1. Wright, S.: Admission Control In Multi-Service IP Networks: A Tutorial. IEEE Communications Surveys & Tutorials, 2nd Quarter (2007)
2. Lima, S., Carvalho, P., Freitas, V.: Admission Control in Multiservice IP Networks: Architectural Issues and Trends. IEEE Communications Magazine (April 2007)
3. Breslau, L., Jamin, S., Shenker, S.: Comments on the Performance of Measurement-Based Admission Control Algorithms. In: IEEE INFOCOM (2000)
4. EU FP7 UniverSelf project website, http://www.univerself-project.eu
5. Eardley, P. (ed.): Pre-Congestion Notification Architecture, Internet RFC 5559 (June 2009)
6. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services, Internet RFC 2475 (December 1998)
7. Menth, M., Lehrieder, F., Briscoe, B., Eardley, P., Moncaster, T., BBabiarz, J., Charny, A., Zhang, X., Taylor, T., Chan, K., Satoh, D., Geib, R., Karagiannis, G.: A Survey of PCN-based Admission Control and Flow Termination. IEEE Communications Surveys and Tutorials (2010)
8. Lundqvist, H., Mas, I., Karlsson, G.: Edge-Based Differentiated Services. In: IEEE IWQoS (2005)
9. Menth, M., Lehrieder, F.: Performance Evaluation of PCN-based Admission Control. In: IWQoS (2008)
10. Menth, M., Lehrieder, F.: Applicability of PCN-based Admission Control, University of Wuzburg Technical Report (2010)
11. Zhang, X., Charny, A.: Performance Evaluation of Pre-Congestion Notification. In: IWQoS 2008 (2008)
12. Latre, S., Vleeschauwer, B., Meerssche, W., Perrault, S., Turck, F., Demeester, P., Schepper, K., Hublet, C., Rogiest, W., Custers, S., Leekwijck, W.: An Autonomic PCN based Admission Control Mechanism for Video Services in Access Networks. In: IFIP/IEEE IM (2009)
13. Roosbroeck, K., Latre, S., Wauters, T., Turck, F.: Optimized Network Utilization through Buffering in PCN enabled Multimedia Access Networks. In: IFIP/IEEE IM (2011)
14. Latre, S., Vleeschauwer, B., Meerssche, W., Schepper, K., Hublet, C., Leekwijck, W., Turck, F.: PCN Based Admission Control for Autonomic Video Quality Differentiation: Design and Evaluation. Journal of Network and Systems Management (2011)
15. Razavi, R., Klien, S., Claussen, H.: A Fuzzy Reinforcement Learning Approach for Self-Optimization of Coverage in LTE Networks. Bell Labs Technical Journal (2010)
16. Dirani, M., Altman, Z.: Self-organizing networks in next generation radio access networks: Application to fractional power control. Elsevier Computer Networks (2010)
17. Chen, G., Pham, T.: Introduction to Fuzzy Sets, Fuzzy Logic and Fuzzy Control Systems. CRC Press (2000)
18. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
19. The NS-2 network simulator, http://www.isi.edu/nsnam/ns/
20. http://www-tkn.ee.tu-berlin.de/research/trace/trace.html
21. Pinson, M., Wolf, S., Stafford, R.: Video Performance Requirements for Tactical Video Applications. In: IEEE Conference on Technologies for Homeland Security (2007)