# Reinforcement Learning for Load Management in DiffServ-MPLS Mobile Networks

Nemanja Vučević, Jordi Pérez-Romero, Oriol Sallent, Ramon Agustí
Dept. TSC, Universitat Politècnica de Catalunya (UPC)
c/ Jordi Girona 1-3, 08034, Barcelona, Spain
{vucevic, jorperez, sallent, ramon}@tsc.upc.edu

*Abstract*—**Cognitive networks are envisaged to provide optimized resource usage in future. While heterogeneity and resource scarcity draw research attention to the wireless part, the rest of the network (mobile backhaul) is rarely considered for these improvements. The future of next generation wireless networks is probable to be all-IP, where a common flexible infrastructure is looking for dynamic autonomous solutions that cognition may provide.**

**This work proposes a novel solution, where the introduction of reinforcement learning over multiprotocol label switching (MPLS) in a differentiated services (DiffServ) mobile backhaul should provide autonomous network adaptation aiming at enhanced QoS capabilities. The proposed solution enables intelligent traffic routing by means of distributed reinforcement learning agents that base decisions on edge-gained experience.**

*Index Terms*—**all-IP, DiffServ, MPLS, QoS, reinforcement learning.**

## I. INTRODUCTION

THE variety of radio access technologies coexisting in time-space domain tend to merge with the objective to provide the constantly growing demands of the users with an adequate QoS ubiquitously. In order to facilitate the inter-technology joint growth, IP technology penetrates in the mobile backhaul [1]. For example the envisaged System Architecture Evolution (SAE) in the long term evolution (LTE) of the 3GPP brings IP technology to the enhanced node B (eNB), whereas the heterogeneous wireless/wired access technologies would rely on a common all-IP backhaul [2].

While the first step in advanced resource usage was to enable heterogeneity in every sense (diversity in radio access technology, services, QoS guarantees, etc.), the further progress starts from the cognitive radio concept [3] where a terminal is supposed to perceive its possibilities in the radio domain and wisely decide on its connecting. The cognition in the radio on the operators' side assumes advanced spectrum management [4] in order to enable optimal usage of the radio resources. Still, the further steps in this sense predict the completely cognitive networks [5][6] where the intelligent autonomous reconfiguration should enable resource utilization with objective to optimize the predefined end-to-end goals.

Most of the work in this field is dedicated to radio resources as the more sensitive network sub-domains, however the increased heterogeneous traffic of wireless users may cause congestion in mobile backhaul as well [7]. With IP common backhaul, the proportions of traffic that enters the network becomes less predictable, raising the need of the operators for the advanced dynamic solutions to handle QoS, avoid unnecessary overprovisioning and preserve low expenditures [2].

In line with the idea of "learn and adjust" to increase the end-to-end network objectives in cognitive networks [8], this work gives a model to introduce cognition in IP mobile backhaul with heterogeneous traffic. The presented solution is developed for the Differentiated Services (DiffServ) [9] domain by means of Multiprotocol Label Switching (MPLS) [10] technology, which becomes a suitable candidate for QoS provision in all-IP networks.

DiffServ, as a QoS provisioning architecture in IP networks, enables class prioritization, resource sharing and service differentiation by different per-hop-behavior (PHB). It defines expedited forwarding (EF) [11] and variants of assured forwarding (AF) [12] as prioritized classes. Unprioritized traffic is classified as best effort (BE). In mobile backhauls mapping of services usually assures that conversational users are mapped onto EF, streaming onto AF and interactive onto lower priority AF or BE [13].

MPLS technology facilitates the use of multiple routes in packet forwarding through a network, by means of packet labeling. The use of multiple paths through a network, unlike the legacy one path routing, enables traffic load balancing when more than one link is able to forward the entering traffic to a desired destination. Load balancing enables wise resource utilization and copes with the expectations of architectures such as SAE that should support many-to-many relations between Access Gateways (aGWs) and eNBs [14]. The need for dynamic autonomously controlled MPLS load balancing is especially identified for the balancing across unequal paths when one path is not offering sufficient capacity, as well as for scenarios with somehow unpredictable traffic loads [15].

When cognitive networks are approached, reinforcement learning (RL) [16] appears as a promising candidate to fulfill the observe-decide-act cycle [6] in network management with simple algorithms. To this end, this work presents a solution to manage load balancing and increase network performances by implementing RL over MPLS in a Diff-

This work has been supported by the Spanish Research Council and FEDER funds under COGNOS grant (ref. TEC 2007-60985); and MEC-FPU Scholarship (AP2005-3739).

Serv environment. Results in this paper prove the capability of the solution to act successfully in situations where unpredictable high priority traffic loads in some nodes may produce bottlenecks, identified as problematic in [15].

Reinforcement learning in MPLS appears rarely in literature (e.g. [17]) and only in [18][19] in combination with DiffServ. Published solutions always rely on traffic engineering, bandwidth brokerage or path reservation as an underlying infrastructure. As a consequence, the entire previous work uses bandwidth estimation or blocking probability as performance indicators. This way the resource management is given an insight on the system capacity on the entire flow path. However, we propose a solution that does not need to be aware of the exact states on the nodes inside an MPLS/DiffServ, but only needs to receive occasional feedback on packet experience from network domain edges. Thus, the proposed solution affects only network extremes. Use of such a domain edge experience for autonomous network adapting by means of RL in scheduling [20] or queue management [21] has been applied to achieve end-to-end goals.

Rest of the paper is organized as follows. Section II gives a description of the proposed solution, followed by section III that explains the evaluation environment and methodology. Results are presented in section IV. Finally, conclusion is given in section V.

## II. Proposed Solution

In this work, we propose a load balancing mechanism for multi-path packet routing in DiffServ environment, by means of MPLS labeling. The decision is based on the experience of packets using different routes to determine the route quality and the probability to use that path further on. To make such a decision, reinforcement learning agents are distributed in routers where multiple path options are available.

The entry and the exit points of an MPLS network are called label edge routers (LERs), which, respectively, push an MPLS label onto the incoming packet and pop it off the outgoing packet. Routers that perform routing based only on the label are called label switch routers (LSRs).

In Fig. 1a, a simple example shows that the flow entering node A may use paths through B and C to reach node D. The ingress LER may split the flow entering it by marking packets in A with appropriate labels. Usual MPLS solutions are based on measuring the bandwidth availability on each of the routers the flows are passing through. However, our solution starts from the information that is taken on node D. The information will reflect the QoS satisfaction level packets experience depending on the path taken, and is therefore measured at the path end. That also means the collected information will be independent on the number of hops (LSRs) between A and D, so B and C may be equivalent to more routers that may appear in corresponding path.

RL needs feedback from the environment to be able to make decisions. This is carried out through the reward function. With this aim, the information on QoS experience is back propagated to RL. This function is likely to be calculated based on the delay and loss measured at point D for the traffic passing through various paths (i.e. between LERs). The exact calculation of reward function is explained in section II.B.

### A. *Ingress LER*

An application of the solution to a simple case is illustrated in Fig. 1a. The traffic entering the node A is marked with different MPLS labels when routed through node B or C. The decision on labeling is made by RL MPLS mechanism, where a RL agent derives probabilities for path selection. Based on those decisions, certain traffic amounts are routed through those paths, and a collective evaluation of the experience on them is looped back to a RL agent that will make further decisions on MPLS labeling.

In Fig. 1b, the example shows three classes case. The traffic is first classified according to the DiffServ class (i.e. PHB) and then the decision of the RL agent is made independently for each class. This approach scales the problem per number of classes that are passing through a domain (one RL agent per class), creating separate virtual domains.

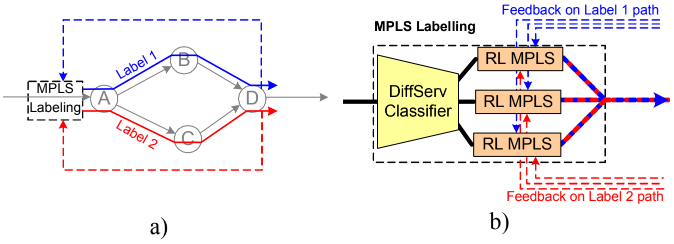


Fig. 1. RL based MPLS labeling: a) two path scenario, b) marking traffic in different DiffServ classes.

### B. *Reward function*

The reinforcement learning agent uses reward function, a feedback from the network, as an evaluation of the previous decisions. This function should reflect satisfaction of the environment the agents are affecting.

In our work, we will use the evaluation of the network behavior in IP network calculated as the following reward function [13]:

$$R_i = \sum_j (c_j \cdot t_{i,j} - p_{loss,j} \cdot l_{i,j} - p_{dly,j} \cdot d_{i,j} - p_{thr,j} \cdot th_{i,j}) \qquad (1)$$

In equation (1) index $j$ stands for traffic class, parameter $c_j$ is the charge per bit for the traffic successfully carried in that class, and $t_{i,j}$ is the amount of traffic (bits) of class $j$ passed through the route $i$. Parameters $p_{loss,j}$, $p_{dly,j}$ are penalizations per packet, for the number of packets lost in transport ($l_{i,j}$) and for those not meeting delay requirements ($d_{i,j}$), respectively. $th_{i,j}$ is the number of activity intervals of a traffic source in which throughput requirements are not accomplished and $p_{thr,j}$ is the corresponding penalization.

### C. *Reinforcement Learning*

The reinforcement learning agent decides on actions to change from one state to another. In our study we consider continuous state-action space while determining the probabilities that the traffic flows take certain routes from a router. The decision on action selection in RL agent bases on experience (i.e. reward feedback) and policies (i.e. RL algorithm). The decision algorithm we use is Softmax Action Selection (SMAS), using Boltzmann distribution [16].

The RL agent updates its decisions after a certain period of time $\Delta T$. For each period RL agent calculates reward using (1), just scaled with the maximum reward assuming neither loss nor penalties are experienced:

$$R_i^{MAX} = \sum_j c_j \cdot t_{i,j}^* \qquad (2)$$

where $t_{i,j}^*$ is the entire input load forwarded through path $i$.

The scaled reward as seen by a RL agent is then:

$$r(i) = \frac{R_i}{R_i^{MAX}} \qquad (3)$$

As we will be tracking a non-stationary problem, the reward will be averaged in time (after each time step $k$) as exponentially weighted (where $\gamma$ is the averaging weight) giving in this way preference to the latest events:

$$Q_{k+1}(i) = Q_k(i) + \gamma \cdot (r_{k+1}(i) - Q_k(i)) \qquad (4)$$

The reward is calculated during time independently for all the possible paths leaving from one router, so choosing one path is seen as action selection by RL agent. Now, at each time instant $k$, the RL agent can use this information and mentioned SMAS criterion to determine the probabilities of choosing action (i.e. path) $a$, over $n$ possible actions (i.e. paths):

$$p_k(a) = \frac{e^{Q_k(a)/\tau}}{\sum_{b=1}^{n} e^{Q_k(b)/\tau}} \qquad (5)$$

Here, $\tau$ is the so called temperature, a positive parameter. The high temperatures cause the actions to be all nearly equiprobable, whereas low temperatures cause a greater difference in selection probability [16].

As explained RL algorithm decides on path selection probability. However, the exact path assignment depends on the flow candidate, and is randomly assigned based on these probabilities.

## III. Evaluation Model

A network model built in OPNET modeler has been used for the evaluation of the proposed solution. The considered network structure is presented in Fig. 2. There are 8 traffic sources (S0-S7), 7 routers (R1-R7), 8 receivers (D0-D7) and two additional load generators in the network. The R1 and R2 are routers where RL algorithm is implemented to decide if traffic is routed through R3 or R4. The evaluation tests the case in which the traffic is passing through a domain that is randomly experiencing congestion depending on load generators in R3 and R4.

### A. *Traffic Sources*

Each source is a traffic generator with a specific class and behavior. Sources from S4-S7 have the same characteristics as sources S0-S3, respectively. Each source generates sessions according to an exponential session interarrival time and constant session duration. Characteristics of sessions are given in Table I. For each randomly derived value the distribution and the mean value are indicated. The described traffic model has previously been used for similar evaluations in [20] and [21].

The specific session interarrival time is varied depending on the total load to be simulated. In Table I the session interarrival times and source bitrates that are given correspond to a case of 100% load. In simulations, the tests consider sources S0-S7 generating load from 50-100% (i.e. $X$% of the load is $X$/100 times the load from table).

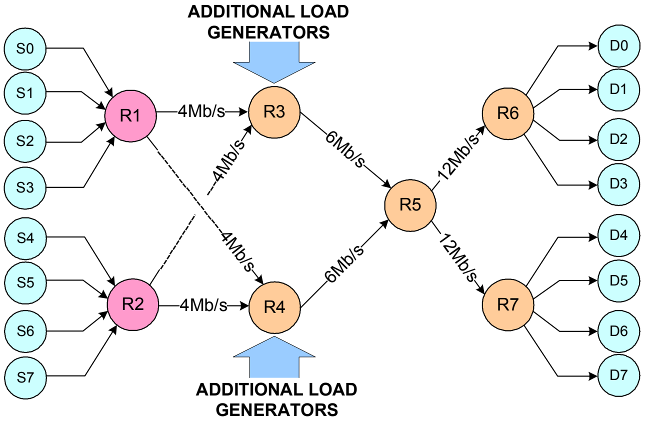


Fig. 2. Network model.

### B. *Routers*

All the routers implement weighted fair queuing (WFQ) [22] as scheduling mechanism to decide the amount of resources allocated to each class. Static weights set as $W_{EF}$:$W_{AF}$:$W_{BE}$=3:2:1 make prioritization to enable increased priority for the classes of higher QoS requirements. The queues are FIFO, with length of 10 for EF, and 100 for AF and BE class, with simple drop from tail mechanism implemented, as in [20]. Note that we use EF/AF/BE to distinguish different classes in this work, however, in general case, these may be AF1/AF4/BE or some other class combinations as well.

The capacities of each link are indicated in Fig. 2.

### C. *Additional Load*

The additional load that is entering the routers is randomly distributed with constant average proportion of classes. Each of the loading flows generates certain constant bitrate (uniformly-distributed random value between 0-4Mb/s for EF class, and 0-1Mb/s for AF and BE traffic) during an interval of random length (exponentially generated value with a 250ms mean independently in each class). In this way the random traffic will have higher proportion if prioritized traffic entering the network, which corresponds to scenarios from [15] for which necessity for autonomous MPLS handling is stressed. The additional traffic load is directed towards R5, and dropped after being received in it.

TABLE I
SOURCE CHARACTERISTICS

| Source | S0, S4 | S1, S5 | S2, S6 | S3, S7 |
|---|---|---|---|---|
| Class | EF | BE | AF | BE |
| Source type | ON/OFF | ON | ON/OFF | ON/OFF |
| Packet interarival | exponential | | | |
| time (s) | 0.015625 | 0.0625 | 0.01333 | 0.03125 |
| Packet size (Bytes) | 125 | 500 | 500 | 500 |
| ON period duration (s) | exp.<br>0.5 | constant<br>1 | exp.<br>0.5 | Pareto<br>0.5 |
| OFF period duration (s) | exp.<br>0.5 | constant<br>0 | exp.<br>0.5 | Pareto<br>0.5 |
| Rate per session in the ON period (kb/s) | 64 | 64 | 300 | 128 |
| Session | exponential | | | |
| interarrival time (s) | 0.96 | 7.68 | 4.5 | 3.84 |
| Session | constant | | | |
| duration (s) | 30 | 60 | 30 | 30 |
| Load (kb/s) | 1000 | 500 | 1000 | 500 |

### D. *Reward Evaluation*

To measure the degree of QoS assurance, the function (1) is used to compute the reward of the overall network. Throughput and all the penalties are calculated on an edge-to-edge basis, for the entire domain.

In EF class all the packets delayed for more than 8ms are penalized. In AF class packets are penalized when delayed for more than 35ms or when equivalent throughput in ON time interval is lower than 200 kb/s. Loss of packets is penalized in all the classes. Parameters in this function are inherited from [20] and are given in Table II.

TABLE II
PARAMETERS OF THE REWARD FUNCTION

| Class (j) | $c_j$ | $p_{loss,j}$ | $p_{dly,j}$ | $p_{thr,j}$ |
|---|---|---|---|---|
| EF | 0.0001 | 0.2 | 0.2 | - |
| AF | 0.00004 | 0.08 | 0.04 | 10 |
| BE | 0.00001 | 0.02 | - | - |

### E. *RL Parameter Setup*

Based on experience and consecutive testing the parameters of the RL agents are set as follows. The averaging parameter is $\gamma$=0.2 for all the classes, whereas the so called temperature parameter is in each class different: $\tau_{EF}$=0.02, $\tau_{AF}$=0.01, $\tau_{BE}$=0.001. Update period is $\Delta$T=10ms.

### F. *Flow Splitting*

In order to achieve load balancing, each flow aggregate that is entering the router must be dividable. In this evaluation study we presume per flow splitting, where each session is distinguishable and assigned a new path. With no loss of generality, session is considered to be new if it has an inactivity period longer than 0.25sec; or after 1sec otherwise.

### G. *Performances Evaluation*

The two routers equipped with RL are having two possible paths occasionally randomly congested (R3 and R4). The RL case will be compared to static case when the MPLS is equally proportioning the balance of each class through the two paths. As the paths are of equal characteristics with equal probabilities to have a bottleneck, this has proven to be the static model that gives best results. Thus this will be the baseline model in this study.

## IV. RESULTS

The purpose of the tests presented here is to show the behavior of the system when routers R1 and R2 in Fig. 2 apply RL algorithms to balance the traffic they input in the network. The RL algorithm will work independently in each router. All the tests were 10000sec long.

The evaluation of the system behavior is done for the cases when the traffic load entering routers R1 and R2 is changing from 50-100% (e.g. 50% load corresponds to half the load values from Table I). For that case the reward function and tracked QoS parameters are compared for the case when reinforcement learning is applied (RL) and for the static case (ST). The resulting behavior in terms of reward function may be observed in Fig. 3, where gain RL case achieves over ST case is marked. As it may be seen gain in reward is present in every class, where most significant contribution is in EF class (~15%). The overall system gain in reward is of ~10.5%.

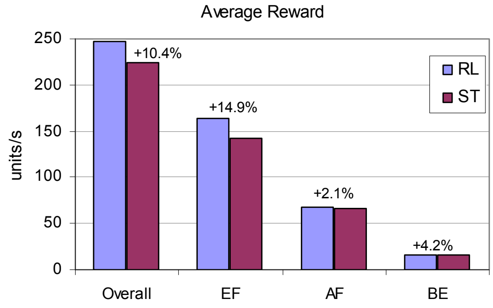


Fig. 3. Average reward for the 100% load case.

As seen, the possibility to sense the network state through the experience based reward enables RL MPLS system to gain significant increase in reward in case of heavily loaded network. However, even in the case of less congested network, when the load entering R1 and R2 is lower, the algorithm may contribute to improve the system behavior. In Fig. 4 the overall network reward is presented when input load varies.

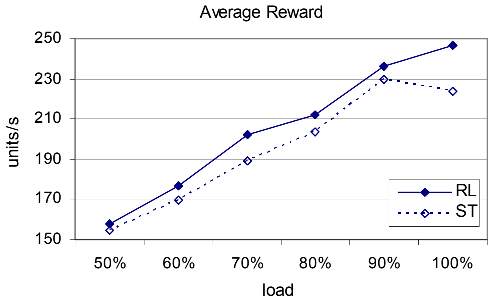


Fig. 4. Average overall reward with varying load amount.

The obtained reward increase when RL agents are controlling network load is a consequence of improvements in tracked QoS parameters. Namely, the reward is calculated as a combination of delay, loss and throughput evaluations. With dynamic path assignment, the RL agent manages to improve these characteristics, or preserve them, in all the cases, as depicted in Fig. 6.

Though the QoS improvements in delay and throughput do not exceed few percents, the most significant improvement lays in fact that these results are achieved while having less loss in each of the classes. In Fig. 5 packet loss ratio may be observed. The statistics show that the loss is lower in all the classes when RL is applied, due to the fact that the agents manage to force alternative paths in case of congestion. From the figure we deduce that the lowest loss is in AF class. The buffer for that class is longer than for the EF class so more packets manage to be preserved in case of conges-

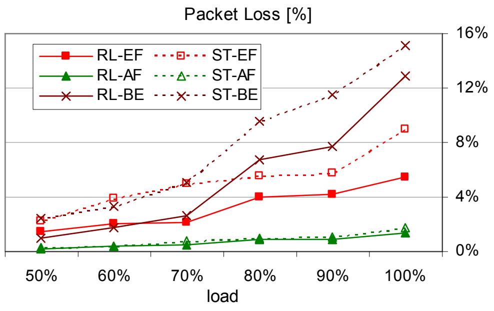
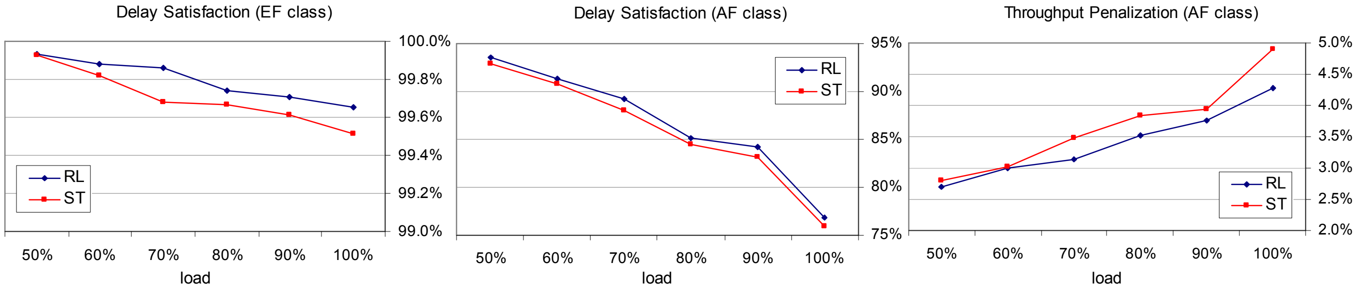


Fig. 5. Packet loss percentage.



a) b) c)

Fig. 6. Percentage of: a) packet with satisfaction of delay limit in EF class; b) satisfaction of delay limit in AF class; c) active intervals with violation of throughput threshold in AF class .

tion. However, this class is more sensitive to delay when load increases, due to its low priority (Fig. 6b).

The previous results confirm the capability of RL agents in the network to learn and act in accordance to traffic needs. On the other hand, looking at the behavior of the RL agents operating independently in the two routers R1 and R2, which are competing for the same resources, results show that taken actions end in a synchronous like behavior. The mutual reinforcement and back-off of the two agents make them produce similar decisions in such a distributed control system, as the experience is affected by the same bottlenecks. This can be observed in Fig. 7 where the probabilities to forward the heterogeneous traffic to R3 from the two agents is presented for the 500sec segment (during an obvious change in network conditions). As it may be perceived both RL agents tend to similar probabilities in corresponding classes.

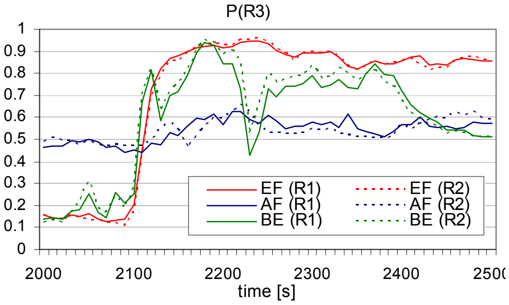


Fig. 7. Probabilities to forward the incoming traffic for all the classes to R3 in R1 and R2.

## V. Conclusion

This work presented a solution for the load management in the mobile IP backhaul based on cognitive concepts. The possibility to introduce reinforcement learning by means of MPLS in a DiffServ environment is presented. The algorithm proved to adapt and provide significant gain to the system under varying network load. The improvements with respect to static case exceed 10%. The improvements in QoS are achieved in all the tracked aspects at the same time. The tests also confirm the possibility to have distributed decision making in independent routers sharing the same resources.

## References

[1] Michael Howard, “Mobile Backhaul Moves to the Forefront”, Infonetics Research, 2006.

[2] 3GPP TR 22.978, *All-IP Network (AIPN) feasibility study*, Jun 2006

[3] J. Mitola, *Cognitive Radio:An Integrated Agent Architecture for Software Defined Radio*, PhD thesis, Royal Inst. of Tech. (KTH), 2000.

[4] P. Leaves, et al, “Dynamic spectrum allocation in composite reconfigurable wireless networks”, *IEEE Communications Magazine*, vol. 42, issue 5, pp. 72-81, May 2004

[5] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, S. Mohanty, “Next Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey”, *ELSEVIER Computer Networks*, vol. 50, Sept. 2006.

[6] R. W. Thomas, L. A. DaSilva, A. B. MacKenzie, “Cognitive networks”, IEEE DySPAN 2005, Nov. 2005

[7] M. Sagfors, V. Virkki, T. Kuningas, “Overload Control of Best-Effort Traffic in the UTRAN Transport Network”, IEEE VTC ‘06, May 2006

[8] R.W. Thomas, D.H. Friend, L.A. DaSilva, A.B. MacKenzie, “Cognitive networks: adaptation and learning to achieve end-to-end performance objectives”, *IEEE Communications Magazine*, vol. 44, issue 12, pp. 51-57, Dec. 2006

[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An Architecture for Differentiated Services*, IETF RFC 2475, December 1998

[10] E. Rosen, A. Viswanathan, R. Callon., *Multiprotocol Label Switching Architecture*, IETF RFC 3031, Jan. 2001

[11] V. Jacobson, K. Nichols, K. Poduri, *An expedited forwarding PHB*, IETF RFC 2598, Jun 1999

[12] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, *Assured forwarding PHB group*, IETF RFC 2597, Jun 1999

[13] Vilho Raisanen, Nokia Networks OY Finland, *Implementing Service Quality in IP Networks*, John Wiley & Sons, 2003

[14] 3GPP TR 23.228, “3GPP System Architecture Evolution – Report on Technical Options and Conclusions”, Nov. 2006

[15] A. Premji, “Using MPLS Auto-bandwidth in MPLS Networks”, Wireline and Wireless Carriers, Juniper Networks, Inc., 2005

[16] R. S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, A Bradford Book, MIT Press, Cambridge, MA 1998

[17] F. Heidari, S, Mannor, L.G. Mason, “Reinforcement Learning - based Load Shared Sequential Routing”, IFIP Networking 2007, Springer LNCS vol. 4479, pp. 832-843, Nov. 2007

[18] C. Scoglio, T. Anjali, J.C. de Oliveira, I.F. Akyildiz, G. Uhl, “TEAM: A traffic engineering automated manager for DiffServ-based MPLS networks”, *IEEE Communications Magazine*, vol. 42, issue 10, pp. 134-145, Oct. 2004

[19] R. Rahim-Amoud, L. Merghem-Boulahia, D. Gaiti, “Autonomous Agents for Self-managed MPLS DiffServ-TE Domain”, Autonomic Networking, Springer LNCS, Volume 4195, Sept. 2006

[20] T. Chee-Kin Hui, C.-H. Tham, “Adaptive Provisioning of Differentiated Services Networks Based on Reinforcement Learning”, *IEEE Transactions on Systems, Man and Cybernetics,* vol. 33, pp. 492-501, Nov. 2003

[21] N. Vucevic, J. Perez-Romero, O. Sallent, R. Agusti, “Reinforcement Learning for Active Queue Management in Mobile All-IP Networks”, IEEE PIMRC ’07, Sept. 2007

[22] A. Demers, S. Keshavt and S. Shenker, “Analysis and Simulation of a Fair Queueing Algorithm”, Proc. SIGCOMM ‘88, vol. 19, Sept. 1989