

A Proposal and Evaluation of a LSP Preemption Policy Implemented with Fuzzy Logic and Genetic Algorithms in a DiffServ/MPLS Test-bed

RODRIGO CAMPOS VIEIRA
PAULO ROBERTO GUARDIEIRO
Federal University of Uberlandia
BRAZIL

Abstract: - The increasing use of multimedia and real time applications in the Internet requires solutions based on resource reservation to provide quality of service. However, the ineffective use of the resources damage such solutions. Thus, the establishment of traffic engineering strategies integrated with proposals of quality of service becomes necessary, aiming to solve this problem. A proposal of solution to such problem uses the preemption to prevent idle resources in the domain searching to take care of other traffics of bigger priority. In this paper, it is proposed and evaluated a LSP preemption politics implemented with fuzzy controller. The validity of the proposal is demonstrated from results obtained in an experimental study.

Keywords: - QoS, DiffServ, Fuzzy Logic Controller, Genetic Algorithms, MPLS, Preemption.

1 Introduction

The currently deployed Internet offers poor support for multimedia and real time applications due to the lack of QoS (quality of service) capabilities. The Differentiated Service (DiffServ) [10] is a proposal that offers QoS assurances, which can aggregate the data flows into different classes of service classifying the flows in the domain edges and producing a scalable solution, however congestion still occurs that can damage its performance. The development of techniques using traffic engineering (TE), as the MPLS (Multiprotocol Label Switching) architecture [6, 18], appears with the objective of simplifying and providing improvements to the network behavior. In the MPLS architecture, a label in each packet that enters in the domain is inserted, so that this packet is analyzed and routed based on its label. The process of forwarding packets through the domain creates the LSP (Label Switched Paths), so that they have similar functions to the virtual circuits of the ATM (Asynchronous Transfer Mode). Therefore, the MPLS by itself isn't able to assure the QoS.

A possibility of integration is the union of DiffServ and MPLS architectures [14], which has been shown sufficiently efficient [9, 7]. An extreme amount of LSP can cause the resources shortage in some links while there are idle resources in the others. This shows a limitation found in this type of integration and that it has been motivated researches. We can find ana-

lytical studies which search for the creation of models and studies based on simulation with the intention of proposing solutions. The studies based on emulation [13] is another possibility that comes aiming at the implementation of a test-bed and the showing experiments with real data.

In this paper, is proposed and evaluated the implementation of a LSP preemption policy. It applies such policy in an emulation test-bed able to implement DiffServ-aware MPLS Traffic Engineering (DS-TE). This test-bed is implemented using conventional PCs and OS Linux. In this scene, a function is specified that shapes how the system can be affected for preemption policy. The following stage is the application of an heuristic search method, genetic algorithms (GA). It realizes the application of rules defined by optimization of preemption politics in the DS-TE domain. These rules change the domain configuration and they modify the LSP creation process with a Fuzzy Logic Controller (FLC) that acts in accordance with traffic behavior.

The organization of this paper is as follows. Section 2 presents related works in the literature. Section 3 gives a problem formulation and the problem solution is proposed. Section 4 shows the tools employed to optimize the QoS in a DiffServ/MPLS environment. Section 5 presents the network environment. Section 6 introduce the results and our comparatives analysis. Finally, in the section 7 the paper is concluded.

2 Related Works

In the bibliographical research regarding the limitation previously cited, the proposal of [11], where is presented the architecture of a management system based on politics that controls the nodes dynamic provisions in a DiffServ domain. Another proposal, the Tequila project [8], uses a signalling protocol that change the domain admission control in agreement with the traffic entrance and in accordance with previously statistical model. Moreover, in [5] a new politics of weighted preemption with an adaptative scheme that minimizes the rerouting is developed. In [17] the MPLS test-bed construction using personal computers (PC) with Operational System (OS) Linux is illustrated and the results to the flows separation are presented . In [12] is presented the development of a MPLS module for installation in OS kernel in order to allow the creation of an MPLS experimental domain.

3 Motivation

3.1 Problem Formulation

Inside a MPLS domain each LSP is created without knowledge about of the available resources given to others LSPs. Thus, when a new LSP is created in a MPLS domain MPLS can cause resource subutilization in the traffic trunks due to non-availability of enough resources to take care of the QoS requirements of the created LSPs (independent of the priority level attributed). This occurs because of the establishing of a traffic trunk to a maximum bandwidth (BW). This is also allocated to be used by this trunk and this limit doesn't take care of the LSP necessities over the whole. This means that the guarantees of the QoS requirements of the multimedia and real time applications will not be reached due to inadequate use of the BW inside of a MPLS domain. Thus, it becomes necessary to implement a mechanism able to manage such resources efficiently.

3.2 Problem Solution Proposal

The question of the arbitrary LSP creation being able to affect the global performance in AS due to the inexistence of available resources is, in truth, a problem of the control class. Due to absence of some control mechanism adequate to the behavior of the system, the waited result isn't reached. In this case, it would be

enough to apply the methodology of classic control on a system model, but such solutions would have as obstacles the uncertainty and inaccuracy found by trying to shape the Internet data flows and the necessity of a qualified professional in the controller adjustment. The first obstacle justifies the use of the FLC due to non-linearity and to the absence of an exact mathematical model for the Internet traffic. Despite this, the second obstacle remains, therefore the FLC needs to be adjusted by a expert. Then, the use of an heuristic search tool (such as the GA) became viable in view of that it's capable of configuring the FLC with optimum parameters. Thus, a LSP prioritization policy is proposed in accordance with the behavior of the data flows in the AS.

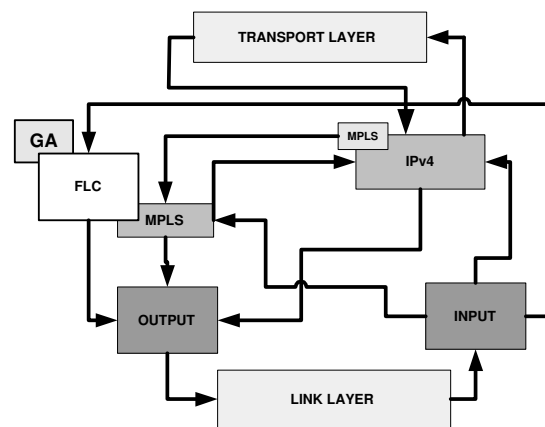


Fig. 1: Proposed network layer implemented

In Fig. 1 the implemented network layer in the test-bed is presented. This Fig. represents the modules gifts in the OS Linux kernel that allows for the network services to be offered. The link layer, the network layer and the transport layer of TCP/IP architecture are shown. The link and transport layers are illustrated in a simplified manner, therefore they don't belong to the target of the work. Moreover, in the figure the network layer is described to show the elements that compose it inside in the OS Linux kernel, such as the IPv4 protocol, the installed MPLS module and the entrance and exit modules. The proposal consists of a FLC module acting as a programmer of MPLS module that determines the politics adopted in the LSP creation. A GA module is also presented which appears as an entity that program the FLC, but because of scalability problems it cannot be implemented in each network node.

4 Development Tools of the Proposal

4.1 Preemption

Preemption is a technique that allows for the optimized resources management in environments where their availability is scarce. In preemptive systems a process can be removed from the CPU (Control Process Unit) due to the arriver of a higher priority process [1]. The objective, in the presented proposal, is to maximize the use of the network resources while the number of connections that would have access to the network denied is minimized.

In [6], the authors present the requirements for use TE in a MPLS domain and they consider the necessity of priority parameters and preemption as TE attributes of a traffic trunk. For this the Class of Traffic (CT) is defined here which represents the set of traffic trunks crossing a link where a specific set of bandwidth constraints is defined. The DS-TE integration can support up to 8 CT ($CT_c, c = 0, \dots, 7$) and each CT is attributed to a bandwidth constraint (BC) or to a constraints set ($BC_b, b = 0, \dots, 7$). According to RDM (Russian Doll Model), in standardization for the IETF [15], and it has been defined that the maximum number of BC is equal to the CT and all LSPs belonging to the CT_c doesn't have to use more than BC_b ($b \leq c \leq 7$ and $BC_b \leq BC_{b-1}$, for $b = 1, \dots, 7$). Also in [6], it had been defined that the two attributes of preemption, s (setup preemption priority) and h (holding preemption priority), which can assume different values and to control the dynamics of the LSP creation. Thus these definitions are used to create a TE-class and to establish preemption policies to act. Such policies can carry through:

- preemption in the connections that have least priority it determines that the QoS of high priority traffics would be better satisfied.
- preemption of the least number of LSPs because the number of LSPs that need to be rerouted would be lower.
- preemption of the least amount of bandwidth that still satisfies the request. In this case, resource utilization would be improved.

The choice of a given politics will determine the composition of the objective function to be optimized so that preemption offers satisfactory response. Being thus, the following objective function is defined in [5]:

$$F(\mathbf{z}) = \alpha(\mathbf{z} \cdot \mathbf{y}^T) + \beta(\mathbf{z} \cdot \mathbf{1}^T) + \gamma(\mathbf{z} \cdot \mathbf{b}^T) \quad (1)$$

where:

\mathbf{b}^T transpose vector with reserved bandwidth.

\mathbf{y}^T transpose vector of holding priority in each LSP.

\mathbf{z} binary vector of preempted LSP.

α, β, γ weights of utilized policies.

Then the employment of the tools becomes necessary to optimize such function for the efficient use of the resources in the domain. In this process, it's necessary to find the global minimum of the function, in accordance with the criteria of the preemption policies established.

4.2 Fuzzy Logic

Fuzzy logic allows one to classify the answers to human experiences what cannot be simply classified as true or false in different levels of truth based on a natural language [4]. Therefore, the fuzzy logic, based in fuzzy set theory, has been adjusted to deal with control problems that possess imperfections in the information as these presented. Thus, fuzzy logic is a tool able to capture imprecise information and to convert it to a numerical format for easy manipulation in computers.

Some studies are being carried out regarding the uncertainty of the network in guaranteeing the QoS of the connections. Works on how [16] demonstrate that when the connection demands only bandwidth, it's more possible to predict the network's behavior than when there are the necessary guarantees of end to end delay and jitter limits, therefore the effect of the network's uncertainty is more perceptible. However, the decomposition of the delay requirements in local constraints and the definition of a distribution probability class becoming the accurate and efficient solution is presented as solution. Problems that involve uncertainty and imprecision finish allowing a solution based on the logic fuzzy, through the FLC implementation [3].

Therefore, a FLC whose pertinence functions establishes in which LSPs must be made preemption in accordance with equation 1 is implemented in the test-bed studied. Such processes could be directly imple-

mented in the FLC, but due to the necessity of scalability and to the dynamic response of the flows behavior in AS, such work is destined to the GA, an heuristical research tool.

4.3 Genetic Algorithms

The genetic algorithms consider the modelling of solutions based on the theory of the evolution and genetics. According to comments of Darwin, the individuals most apt to survive in a certain environment have the characteristic to incorporate the adequacies used for survival to its genetic information. and they transmit it through many generations of a population. Thus, the genetic algorithms propose the creation of a population, which represents a candidates set to the solution of the evaluated problem. These individuals are selected in accordance with its fitness and generate a new population, through genetic operations [2]. The process is iterative until it determines the qualified individuals to supply an excellent solution in accordance with its fitness. There are many applications using genetic algorithms, but the integration of this technique for one training of fuzzy systems through the heuristical search of optimum solutions has deserved prominence.

As was presented previously, a cost function $F(\mathbf{z})$ is responsible for adjusting the minimization of the variable \mathbf{z} , which is shaped here as a chromosome of the population and that it can suffer variations. Thus, the chromosome is composed for the vector \mathbf{z} , for the adjusted global cost calculated through the function, for the average fitness and the general fitness of the individual in relation to the population. The algorithm is controlled for the number of iterations, being observed that the mark of fifty iterations has been enough for the stability of the results.

5 Test-bed Implementation

5.1 DiffServ/MPLS Test-bed

Considering the proposed methodologies for the QoS providing in IP networks, it was opted the use of a DiffServ/MPLS integration aiming at getting a differentiation of services of efficient way and a reduction of the end to end delay tried in conventional IP networks. Then, a simplified test-bed was used (Fig. 2), which allows to shape in a simplified way an AS of the Internet. For the construction of this test-bed it had used

conventional PCs and OS Linux, with the purpose to implement routers. The choice of these elements allows a generic study, as well as the cost minimization.

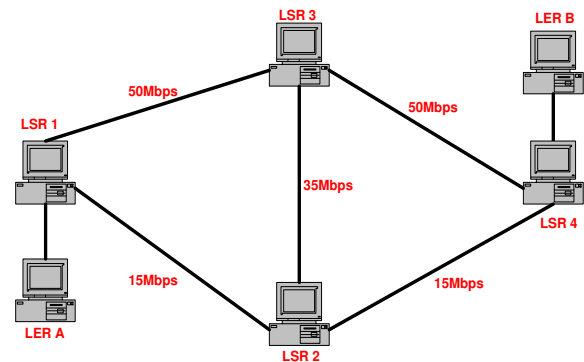


Fig. 2: Test-bed studied

As one can observe in Fig. 2, the links had been configured to present distinct bandwidths, forming the standard routing benefits as those with high capacity, independently to have or not congestion. Thus, in the upper branch 50 Mbps were allocated and 15 Mbps in the under branch.

6 Results Presentation and Analysis

In this section is analyzed the test-bed under different traffic conditions and, later, the methodologies of artificial intelligence are applied to reach the LSPs prioritization obtained with preemption. At a first moment it's necessary to observe the behavior of the test-bed when subject to common traffic sources using TCP and UDP protocols, thus being distinguished as a assimilation, on average, raised to the link capacity it presented the application using UDP protocol.

Fig. 3 proves that applications that use this protocol, generally shaped as CBR sources CBR and representing real time traffic, need a well planned attendance, otherwise it can compromise the others applications. In this scene, the application of mechanisms of flow differentiation becomes necessary, such as the DiffServ, to make this task.

In Fig.s 4 and 5 one observes the non-differentiation of services provided by this environment, so that the same doesn't possess activated mechanisms that make this procedure possible. It must be considered, then, that in accordance with the type of application, this behavior of delay becomes undesirable, compromising the correct attendance to the ser-

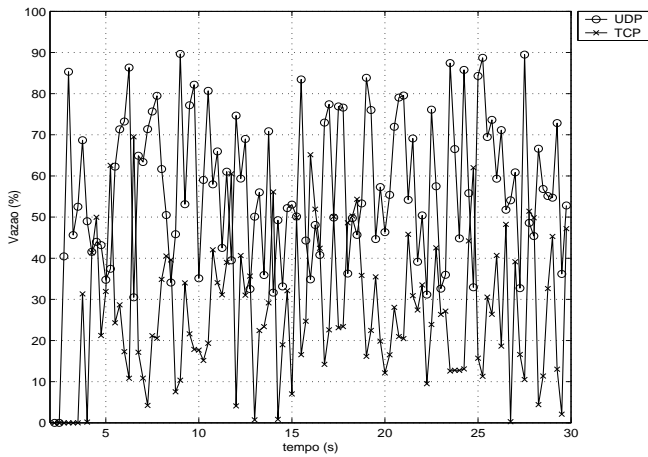


Fig. 3: Best-Effort traffic throughput

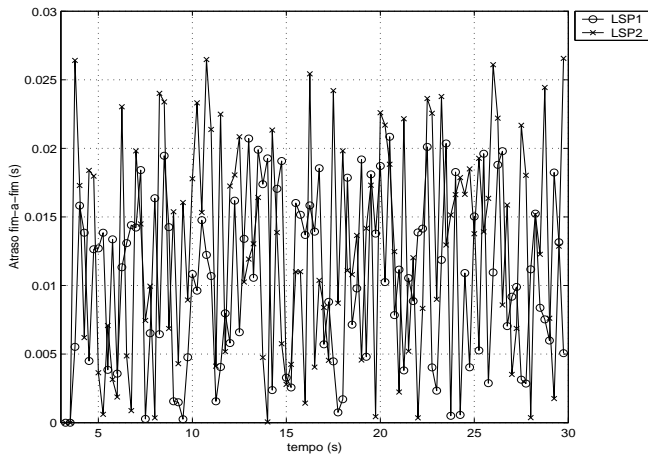


Fig. 4: Best-Effort traffic end to end delay

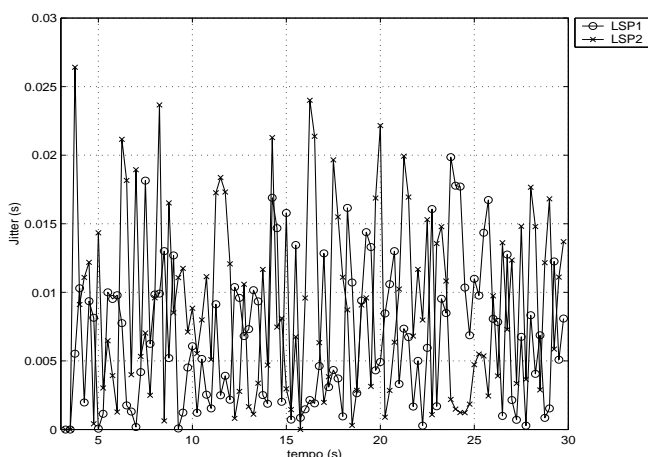


Fig. 5: Best-Effort traffic jitter

vices. The Fig. 5 (jitter, or variation of the delay) also illustrates this problem showing the possible adversi-

ties that real time applications face in a common AS.

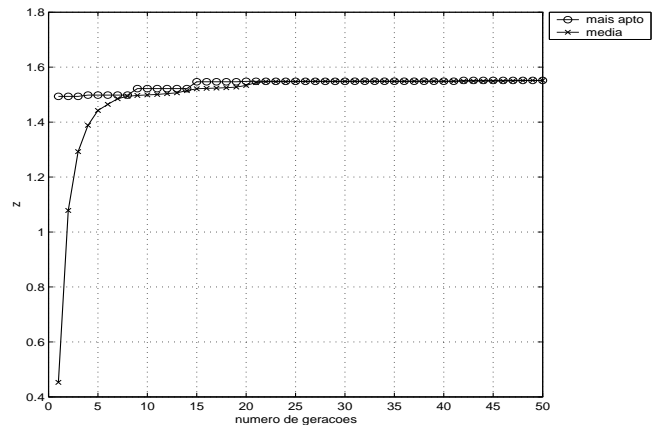


Fig. 6: GA performance curve

The non-differentiation of services observed in the Fig. 3 has motivated many researchers to consider solutions, such as the DiffServ. Moreover, the existence of some limitations of new connections was also observed, a fact that motivated the development of policies of preemption in a MPLS environment and the integration with DiffServ solution. However, the mechanism that executes the control of its performance is interesting, moreover if it turns out to be an intelligent mechanism as the FLC supported by a GA.

Fig. 6 illustrates the performance of the genetic algorithm for a simulation with parameters of the study environment (the behavior of the optimum parameters and the average of the population of parameters derived from the pertinence function of the FLC with scale adjustment). It had been observed that the chromosome most able to minimize the function was obtained in a relatively small number of generations, considering itself that the execution time is linked to the accuracy of the algorithm and the choice of the appropriate language.

Fig. 7 illustrates the obtained result being attributed to LSP3 flow high priority, also on the LSP1, taking it to appropriate larger amount of LSPs and to use the bandwidth as was used for LSP2 traffic. One needs notice to that, practically in the instant that the source receives resources and initiates the information transmission in AS, LSP3 appropriates to the bandwidth of LSP2 traffic. This traffic was disabled to send packets and tried high indices of loss. Therefore, LSP3 traffic used the best way to link capacity and it finished getting throughput larger than the traffic belonging to

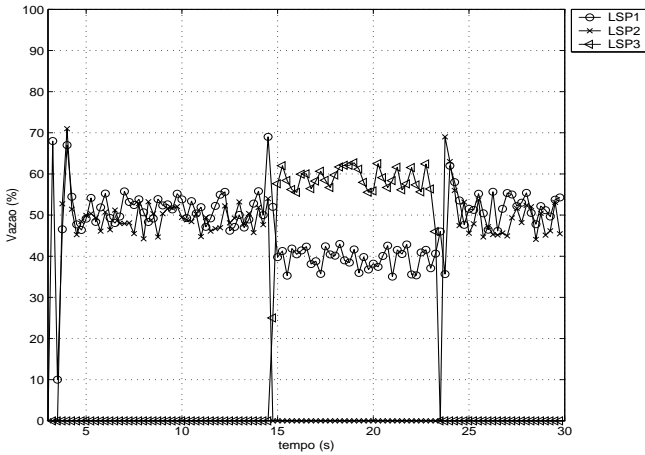


Fig. 7: Preemptive test-bed throughput

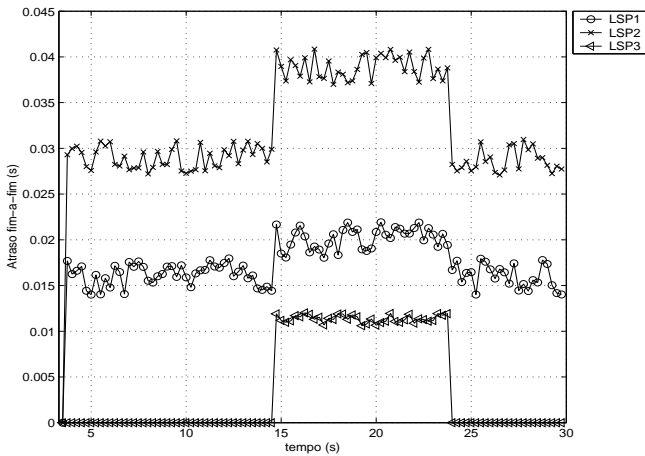


Fig. 8: Preemptive test-bed end to end delay

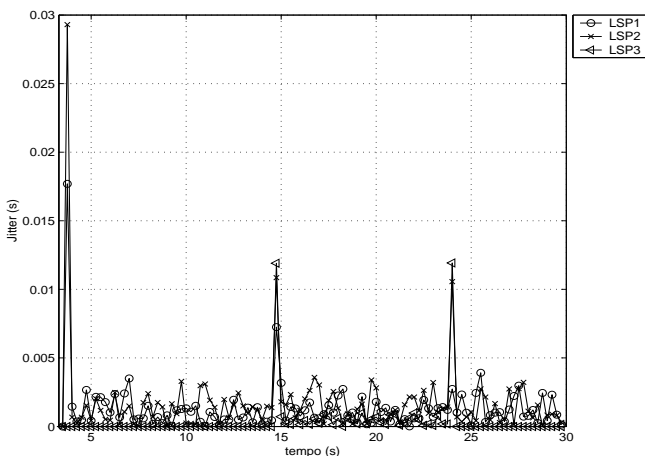


Fig. 9: Preemptive test-bed jitter

the LSP1 in this interval of time.

Moreover, it perceives that the end to end delay

and the jitter, essentially important parameters for real time applications, obtain a reduction in its values and it start to have a behavior less subject to variations, as it's observed in the Fig. 8 and Fig. 9. This allows to conclude that the proposed method of preemption possess applicability, but knowledge of the behavior of the traffic in the AS is necessary to determine in which situations the preemption policies must be used.

7 Conclusion

In this paper is presented the proposal and the evaluation of a LSP preemption policy using a FLC and a genetic algorithm as a support. This is made in a DiffServ/MPLS test-bed, which is implemented with the use of an open architecture (free software and conventional hardware). Initially, this test-bed representing an AS was used for the accomplishment of experiments considering the Best Effort model. The results obtained in terms of throughput, end to end delay and jitter had been checking the no differentiation of services. Later, the test-bed was configured to carry through the DiffServ/MPLS integration. In this case there were also used in AS the preemption policies configured according to the lesser amount of LSPs or lesser amount of bandwidth placed for LSP criteria. These criteria had been shaped as a function that served as search object to proposed genetic algorithm to find a best value to minimize it.

The obtained results had demonstrated that if we attribute greater priority to a flow in a LSP than the priority attributed to others, this appropriates the necessary amount of bandwidth to take care of to its requirements. It was also noted that, in the instant of the flow creation of this new LSP an increase in the end to end delay tried at for other LSPs occurred due to the priority given to this new LSP. Therefore this scene becomes interesting from the point of view of applications that need attendance with time restrictions. Thus the implementation of preemptive policies assisted by intelligent control mechanisms (FLC), proposal and evaluated in this article, become perfectly viable. However, its use is recommended for applications that need the resources for a short interval of time. This occurs because the other flows of data will always be damage when obtaining the attendance of the QoS requirements of interactive multimedia and real time applications.

Acknowledgement

The authors gratefully acknowledge the financial support of FAPEMIG under Project TEC-1027/02.

References:

- [1] T. P. Baker. *Stack-Based Scheduling of Real-Time Processes*. IEEE Computer Society Press, 1993.
- [2] D. Beasley, D. R. Bull, and R. R. Martin. A overview of genetic algorithms:part 1. *University Computing*, pages 58–69, 1993.
- [3] R. Cheng and C. Chang. Design of a fuzzy traffic controller for atm networks. *ACM Transactions on Networking*, 4:460–469, June 1996.
- [4] R. de Caluwe. *Fuzzy and uncertain object-oriented databases: concepts and models*. British Library, Singapore, 1997.
- [5] J. C. de Oliveira. *New Techniques for End-to-End Quality of Service Provisioning in Diff-serv/MPLS Networks*. PhD thesis, Georgia Institute of Technology, March 2003.
- [6] D. Awduche et al. Requirements for traffic engineering over mpls. Internet RFC 2702, September 1999.
- [7] J. F. Chiu et al. Supporting end-to end qos in diffserv/mpls networks. *10th ICT*, 1:261–266, 23 February-1 March 2003.
- [8] P. Trimintzios et al. An architectural framework for providing qos in ip diferential service networks. *VII IFIP/IEEE*, 2001.
- [9] S. Avalone et al. An experimental analysis of diffserv-mpls interoperability. *10th ICT*, 1:281–287, 23 February-1 March 2003.
- [10] S. Blake et al. An architecture for differentiated services. Internet RFC 2475, December 1998.
- [11] M. Fernandez, A. Pedroza, and J. Rezende. Optimizing fuzzy controllers with genetic algorithms for qos improvement. *ITS2002*, 2002.
- [12] P.V. Heuven, S.V. Den-Berghe, and J. Coppens. Rsvp-te daemon for diffserv over mpls under linux. Atlantis Project, 2004.
- [13] J.R. Leu and R. Casellas. Project mpls for linux. Sourceforge Project, 2004.
- [14] F. le Faucher et al. Multiprotocol label switching (mpls) support of differentiated services. Internet RFC 3270, Maio 2002.
- [15] F. le Faucher et al. Russian dolls bandwidth constraints model for diffserv-aware mpls traffic engineering. Internet draft, draft-ietf-tewg-diff-te-russian-02.txt, Maro 2003.
- [16] D. Lorenz and G. Orda. Qos routing in networks with uncertain parameters. *Infocom 98*, 1998.
- [17] G. Rosenbaum, S. Jha, and M. Hassan. Empirical study of traffic trunking in linux-based mpls test-bed. *International Journal of Network Management*, 13(4), July-August 2003.
- [18] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. Internet RFC 3031, January 2001.