Self-similar QoS Manager: a bio-inspired approach

Alessandro Vivas, Luciana Assis
Computing Department
UFVJM - Diamantina, Brasil
e-mail: alessandrovivas,lupassis@ufvjm.edu.br

Luciano de Errico
Electronic Engineering Department
UFMG - Belo Horizonte, Brasil
e-mail: l.errico@cpdee.ufmg.br

Abstract— **This work presents a Self-similar QoS Manager using a immunological algorithm to optimize a MPLS network. The system receives a set of requirements, imposed by many different client applications, and creates a set of paths that satisfy all the capacity constraints necessary for each specific traffic. The solution also keeps the load balance in the network and considers the self-similar characteristics of the traffic, which lead to improved results. The ns2 (network simulator 2), working with real traffic traces injected in source nodes, was used to validate the approach. Results showed that all simultaneous client application requirements were satisfied, while optimizing the traffic allocation in the MPLS network. The paper also analyzes the influence of different bio-inspired algorithms (genetic and immunological) in the solution of the MPLS path allocation problem.**

Keywords*— *QoS; MPLS path allocation problem; traffic engineering; self-similar traffic; bio-inspired optimization.

I. Introduction

This paper aims to develop a Self-similar Qos Manager using bio-inspired algorithms. This system receives multiple simultaneous requests to create optimized paths, depending upon requests placed by clients and constraints imposed by the network topology. This paper presents an algorithm to statically solve the MPLS path allocation problem for self-similar traffic in networks with capacity constraints.

Multiprotocol Label Switching (MPLS) is a popular routing technique for IP networks where the core problem is to find a route (called LSP) that satisfies all capacity constraints imposed by a specific traffic. MPLS consists in routing packets based on a label, which is inserted in every packet between the corresponding link and network headers. Reduction of IP packets routing process complexity and traffic engineering introduction are two of the main motivations for using MPLS [1].

Genetic algorithms are search and combinatory optimization methods based on natural selection conceived by Charles Robert Darwin. Natural selection states that the most adapted generation remains, while the less adapted disappear with time. Genetic algorithms are evolutionary algorithms, which initially consider an initial population and evolve through the genetic operators of selection, crossover, and mutation.

A genetic algorithm can be defined as a kind of biased random search technique, developed by Holland [2], able to get solutions in a complex multidimensional space. One of the advantages of genetic algorithms is that they deal with a population of simultaneous points, selecting the best ones, making possible to create a subset from the original population not only near the global solution but also in other regions of the search space.

Evolution in a given population happens when selection, crossover and mutation operators are applied to several generations. These methods affect the success of a genetic algorithm and the associated effects can vary according to the kind of the problem [3]. Factors such as crossover and mutation rates, population sizes and elitism techniques must be evaluated when a genetic algorithm is used to solve a given problem.

Artificial immune systems (AIS) are computational systems inspired by the principles and processes of the vertebrate immune system. These kinds of algorithms exploit characteristics like learning, memory and somatic hypermutation. Copt-ainet is an immunological algorithm used to deal with combinatorial problems [4] and use principles of affinity maturation [5]. Two local searches procedures were performed to work with affinity maturation (Tabu Search and Grasp).

II. System Details

The Self-similar QoS Manager was implemented in C language and starts its execution reading a file with information about network topology and LSP requests. Finishing execution, the Manager generates scripts for the ns2 simulator, allowing a faster evaluation of the results produced by the optimization process. The information about LSP requests comprises: number of requests, source and destination node, throughput flow (Mbps), maximum delay allowed (ms), delay jitter (ms), maximum packet loss (%), average packet size (bytes), traffic variance (σ), Hurst exponent (H) and traffic type.

For each sent requisition, k-shortest paths are determined using the algorithm presents in [6], [7]. K-shortest paths are known to be possible solutions for the problem, i.e., finding LSPs. Binary encoding has been adopted for the definition of individuals, where each possible path was represented by a binary value.

In the present work, it has been used the mathematical formulation proposed by Girish [8]. This formulation contains the following statements: routers are designated as LSRs; U is the set of source and destination LSRs; F is the LSP set; and D is a set of associated demands to F. However the mathematical model has the following parameters: u_l - LSR source; v_l - LSR destination; μ_l - bandwidth; e a_l - link cost. The decision variable x_{ij} shows whether the LSP will be routed through the link:

$$x_{ij} = \left\{ \begin{array}{ll} 1, & \text{if the LSP } i \in F \text{ is routed to the link } l \in E \\ 0, & \text{otherwise} \end{array} \right\} \quad (1)$$

The model is given by

$$Minimize(Z_R) = \sum_{l \in E} a_l \sum_{i \in F} \lambda_i x_{il}, \quad (2)$$

submitted to

$$\sum_{i \in F} \lambda_i x_{il} \le \mu_l, \qquad \forall l \in E \quad (3)$$

$$\sum_{\forall l | u_l = n} x_{il} = 1, \qquad \forall n \in U \forall i \mid s_i = n \quad (4)$$

$$\sum_{\forall l | v_l = n} x_{il} = 1, \qquad \forall n \in U \forall i \mid d_i = n \quad (5)$$

$$\sum_{\forall l | u_l = n} x_{il} - \sum_{\forall l | v_l = n} x_{il} = 0, \qquad \forall n \in Vi \mid s_i \ne n, d_i \ne n \quad (6)$$

$$x_{il} \in \{0,1\}, \qquad \forall i \in F, l \in E \quad (7)$$

The constraint (3) assures that the link capacity will not be exceeded. The constraints (4) and (5) assure to all LSPs that start and terminate in a LSR are routed. The constraint (6) assures that every LSP is routed through intermediate nodes. The constraint (7) specifies that each decision variable will assume 0 or 1. The objective function can be calculated using the algorithm presented in Algorithm 1.

Algorithm 1: Objective Function

1 **for** *each link in network topology compute* **do**
2   The LSPs to each link;
3   The mean packet size;
4   Aggregate variance (Equation 9);
5   Aggregate Hurst Exponent (Equation 8);
6   Equivalent capacity (Equation 10);
7   Residual capacity (Total capacity minus alocated capacity);
8   Penalty function (Equation 11);
9   Delay (Equation 13);
10 **end**
11 Total cost to alocate all LSPs;
12 **for** *all LSPs* **do**
13   Compute the delay for each LSP (sum of delay of all links passed by LSP) ;
14   **if** *LSP_i delay > LSP_i delay required by application* **then**
15   Apply a penalty ;
16   **end**
17   Update the total penalty;
18 **end**
19 Compute the total fitness the individuals.;

Values of the aggregate Hurst exponent and variance are calculated in steps 3 and 4. This concept was proposed by [9] and estimated Hurst exponent and variance depending on traffic aggregation. In this model, the aggregate traffic is $A_{ag}(t) = \sum_s A_g(t)$, where A_g is a source of self-similar traffic, H is the Hurst exponent and σ^2 is the traffic variance. The mean traffic is computed by the average of all traffic sources, Equation 8 computes the Hurst exponent, and Equation 9 computes the aggregate variance.

$$H_{ag} = \sum_s H_s \sigma_s^2 / \sum_s \sigma_s^2 \quad (8)$$

$$\sigma_{ag} = \sum_s \sigma^2 \quad (9)$$

The effective capacity is calculated in step 6 (Equation 10). Effective bandwidth formula is useful for estimating transmission capacity required to support network traffic [10].

$$\hat{c} = a + K^{\frac{H-1}{H}} (k\sigma)^{1/H} H (1-H)^{\frac{H-1}{H}} \quad (10)$$

Where $\hat{c}$ is the link capacity that an overflow in K size buffer can occur with ε probability [11].

Step 7 computes the penalty to the load balance function. In this work we have used the cost function proposed by Fortz and Throup [12]. The set of links is A, an arc is a, load is l_a and c_a is the capacity of link. Each $a \, \varepsilon \, A$ had a cost function associated with link occupation ($\frac{l_a}{c_a}$). Function ϕ computes the total cost of allocation.

$$\phi = \sum_{a \varepsilon A} \phi_a(l_a) \quad (11)$$

where $\phi_a(l_a)$ is

$$\phi_{la} = \left\{ \begin{array}{lllllll} 1 & \text{to} & 0 & \le & l_a/c_a & < & 1/3, \\ 3 & \text{to} & 1/3 & \le & l_a/c_a & < & 2/3, \\ 10 & \text{to} & 2/3 & \le & l_a/c_a & < & 9/10, \\ 70 & \text{to} & 9/10 & \le & l_a/c_a & < & 1, \\ 500 & \text{to} & 1 & \le & l_a/c_a & < & 11/10, \\ 5000 & \text{to} & 11/10 & \le & l_a/c_a & < & \infty \end{array} \right.$$

In the step 8, delay is computed by envelop process. The envelope process is an upper bound to the volume of arrivals from a multi-fractal Brownian motion [11]. Equation 12 computes q_{max}, where q_{max} is the maximum buffer size in a time scale, where c is the effective capacity, H is the Hurst exponent, a is a node traffic, σ is the traffic variance, and K is the buffer size.

$$q_{max} = (c-a)^{\frac{H}{H-1}} (k\sigma)^{\frac{1}{1-H}} H^{\frac{H}{1-H}} (1-H) \quad (12)$$

The delay node can be calculated by Equation 13 [13], where d is delay, q_{max} is calculated in Equation 12 and c is effective capacity computed by Equation 10.

$$d = \frac{q_{max}}{c} \quad (13)$$

Loss rate is computed using the Equation 15 where t_{max} is computed in Equation 14 [14].

$$t_{max} = \left[\frac{k\sigma H}{(c-\bar{a})} \right]^{\frac{1}{H-1}} \tag{14}$$

$$P_{max} = \frac{\hat{A}_H(t_{max}) - Ct_{max} - B_f}{\hat{A}_H(t_{max})} \tag{15}$$

III. Experimental Results

In order to standardize tests, it was established two minimal paths for every LSP and all the topologies were tested with a number of requisitions varying from 50 to 500 LSPs. Considering 50 LSPs and two possible minimal paths for each LSP, we have a chromosome of 50 bit-long, which implies in 2^{50} combinations in the search space. For 500 LSPs, the search space is 2^{500} combinations.

In [15] a genetic algorithm is applied to the same model of Equation 2. This paper studied the influence of different crossover and selection methods in achieving a fast and accurate convergence of the genetic algorithm, when solving the MPLS allocation problem. Experimental results, using different network topologies, such as Carrier, Dora, and Mesh, have shown that uniform crossover and Stochastic Remainder Sampling (SRS) selection are the most suitable combination to solve the problem.

In [16] Copt-Ainet and Opt-Ainet immunologic algorithms were applied in the path-finding problem for MPLS networks. Improvements were introduced in the Copt-Ainet algorithm during the affinity maturation stages in order to increase performance. In sequence, both algorithms were compared to a Genetic Algorithm (GA) and it was observed the superiority of the immunologic algorithms for the evaluated network scenarios, with a high number of routes. In contrast, for networks with a small number of routes, the AG presented a more interesting approach, given its smaller execution time if compared to the implemented immunologic algorithms.

In the present work, the best genetic algorithm found in [15] and the Copt-ainet algorithm found in [16] were applied. The Copt-ainet was superior to the Genetic Algorithm, reducing the total cost of allocation in 15%.

To validate the optimization model a simulation with ns-2 was executed using the results of the Copt-ainet algorithm. The simulation used 20 traffic sources of 1.1 Mbps, variance of $269,36x10^9$, Hurst exponent of $0,80$, and maximum delay of 10 ms for each LSP. The topology is shown in Figure 1. This network has 6 nodes (LSRs), 20 sources, 20 destination nodes, and all links with 10 Mbps and delay of 1 ms. The input traffic in the source nodes is a real traffic trace of Bellcore Morristown Research and Engineering Center.

Delay values, throughput, delay jitter, and packet loss were collected for two different simulations (MPLS IP network and IP only network, without MPLS). Delay and delay jitter for the two simulations are presented in Table 1.

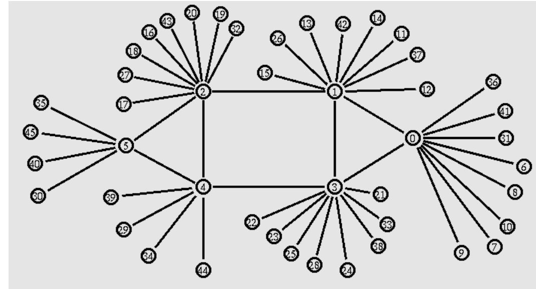

Fig. 1. Self-similar Simulation

Delay was reduced by 20.18% and delay jitter reduction was 33.61%, when the allocation system was used. Throughput was the same for both simulations and there was no observed packet loss.

TABLE I
RESULTS OF SIMULATION

	Delay [s]		Jitter [s]	
LSP	MPLS	IP	MPLS	IP
1	4.62E-003	4.85E-003	4.10E-004	4.49E-004
2	7.02E-003	7.02E-003	5.97E-004	5.97E-004
3	5.29E-003	5.29E-003	4.54E-004	4.54E-004
4	6.88E-003	6.88E-003	6.21E-004	6.21E-004
5	4.60E-003	4.61E-003	3.93E-004	3.94E-004
6	4.15E-003	4.25E-003	3.17E-004	3.50E-004
7	4.82E-003	4.82E-003	3.56E-004	3.56E-004
8	4.82E-003	4.82E-003	3.63E-004	3.63E-004
9	7.55E-003	8.33E-003	7.49E-004	8.16E-004
10	6.42E-003	5.77E-003	5.55E-004	5.05E-004
11	8.01E-003	7.34E-003	6.22E-004	5.55E-004
12	7.66E-003	8.62E-003	6.84E-004	7.81E-004
13	5.09E-003	4.71E-003	4.65E-004	4.17E-004
14	8.49E-003	7.47E-003	6.81E-004	6.03E-004
15	9.29E-003	9.72E-003	8.48E-004	9.36E-004
16	5.24E-003	5.24E-003	4.02E-004	4.02E-004
17	6.01E-003	6.76E-003	5.69E-004	6.42E-004
18	5.22E-003	5.22E-003	3.86E-004	3.86E-004
19	7.92E-003	8.74E-003	7.24E-004	7.84E-004
20	7.11E-003	7.90E-003	7.21E-004	8.25E-004

IV. Conclusion

This work presents a Self-similar QoS Manager based on bio-inspired algorithms. This system allows the optimized allocation of LSPs (MPLS paths) to satisfy sources with self-similar traffic, keeping the load balance and following constraints imposed by the network topology and client applications. In the present version the system works in a static way, i.e., all requests are known before the start of the execution.

One contribution is the study of the application of bio-inspired heuristics for such problems. The implemented Copt-ainet algorithm showed results 15% superior to the Genetic Algorithm for the same NP complete problem. Another contribution is the different approach. The Hybrid

Traffic Model was proposed in [17], using a combination of fBm (fractional Brownian motion) and Markovian traffic characterization procedures, combined with link parameters. In this paper, the mathematical model proposed in [8] was combined with load balance equations [12] for self-similar traffic. The obtained results show the importance of considering the traffic model while developing traffic engineering systems. Fail to use self-similar traffic equations leads to an underestimation of the required bandwidth, which increases packet loss and does not guarantee QoS parameters to customers. Additionally, in a practical point of view, the developed framework also enabled a seamless integration with the simulator ns2, which reduced the time for preparing the simulations.

For future works, one possible improvement is to modify the system to work dynamically, again in integration with the simulator. Work on the improvement of the bio-inspired core is another path to be followed, experimenting with different genetic and immunological algorithms.

V. Acnowledgments

The authors would like to thank the FAPEMIG and CNPq for their financial support.

References

[1] D. O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over mpls," Internet Engineering Task Force, Freemont, RFC 2702, Sep. 1999. [Online]. Available: \url{http://www.rfc-editor.org/rfc/rfc2702.txt}

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1992.

[3] J. A. Vasconcelos, J. A. Ramirez, R. H. Takahashi, and R. R. Saldanha, "Improvements in genetic algorithms," *In Transactions on Magnetics*, vol. 37, pp. 3414–3417, 2001.

[4] J. S. de Souza, F. J. V. Zuben, L. N. de Castro Silva, and J. S. Sousa, "An immune-evolutionary algorithm for multiple rearrangements of gene expression data," in *Genetic Programming and Evolvable Machines*, vol. 5. Hingham, MA, USA: Kluwer Academic Publishers, 2004, pp. 157–179.

[5] L. C. T. Gomes, F. J. V. Zuben, G. B. Bezerra, and L. N. Castro, "Copt-ainet and the gene ordering problem," *Revista tecnologia da informao*, vol. 3, no. 2, pp. 27–33, 2003.

[6] E. Minieka, "On computing sets of shortest paths in a graph," *ACM Communications*, vol. 17, no. 6, pp. 351–353, 1974.

[7] E. Martins, M. Pascoal, and J. Santos, "The k shortest paths problem," CISUC, Coimbra, Portugal, Tech. Rep., June 1998.

[8] M. K. Girish, B. Zhou, and J.-Q. Hu, "Formulation of the traffic engineering problems in mpls based ip networks," in *ISCC Proceedings of the Fifth IEEE Symposium on Computer and Communications (ISCC 2000)*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 214–219.

[9] P. Orenstein, H. Kim, and C. Lau, "Bandwidth allocation for self-similar traffic consisting of multiple traffic classes with distinct characteristics," *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 4, pp. 2576–2580 vol.4, 2001.

[10] A. A. Patel and C. L. Williamson, "Effective bandwidth of self-similar traffic sources: Theoretical and simulation results," in *Proceedings of the IASTED Conference on Applied Modeling and Simulation*. Banff, Canada: IEEE, July 1997, pp. 298–302.

[11] N. L. S. Fonseca, G. S. Mayor, and C. A. V. Neto, "On the equivalent bandwith of self-similar sources," *ACM Transactions on Modeling and Computer Simulation*, vol. 10, no. 2, pp. 104–124, 2000.

[12] B. Fortz and M. Thorup, "Increasing internet capacity using local search," *Comput. Optim. Appl.*, vol. 29, no. 1, pp. 13–48, 2004.

[13] K. J. Ng, B. Tang, and W. Zhang, "The statistical end-to-end delay guarantee for networks with self-similar traffic," Department of Computer Science, Baptist University, Hong Kong, China, Hong Kong, China, Tech. Rep., 2002.

[14] P. A. S. M. Barreto, "Uma metodologia de engenharia de trfego baseada na abordagem auto-similar para a caracterizao de parmetros e a otimizao de redes multimdia," Ph.D. dissertation, Universidade de Braslia, 2007.

[15] A. Vivas, L. de Errico, A. L. L. Aquino, L. Assis, and C. H. N. R. Barbosa, "Analysis of selection and crossover methods used by genetic algorithm-based heuristic to solve the lsp allocation problem in mpls networks under capacity constraints," in *EngOpt 2008 - International Conference on Engineering Optimization*, 2008.

[16] A. Vivas, L. de Errico, and L. Assis, "Estudo comparativo entre algoritmos imunolgicos e algoritmo gentico aplicados ao problema de estabelecimento de rotas com restrio de capacidade utilizando o protocolo mpls," in *Simpsio Brasileiro de Pesquisa Operacional*, 2009.

[17] P. S. Barreto and P. H. P. de Carvalho, "Hybrid traffic model for multimedia network performance evaluation," in *AINA Workshops*, 2008.