A Traffic Engineering proposal for ITU-T NGNs using Hybrid Genetic Algorithms

Alex Vallejo[1], Agustín Zaballos[1], David Vernet[1], Albert Orriols-Puig[1] and Jordi Dalmau[2]

[1] Enginyeria i Arquitectura La Salle
Universitat Ramon Llull (URL)
Barcelona, Spain
{avallejo, zaballos, dave, aorriols}@salle.url.edu

[2] Abertis
Barcelona, Spain
jordi.dalmau@abertistelecom.com

***Abstract*—Routing optimization is a key aspect to take into account when providing QoS in next generation networks (NGN), especially in access networks. The problem of weight setting with conventional link state routing protocols for routing optimization has been studied in order to adjust link's utilization and it has been object of study by a few authors. Among different approaches, GAs have been devised as one of the most appealing methodologies to tackle this problem since it becomes NP-hard when applied to large networks. In particular, some authors have used hybrid GAs (memetic GAs) which incorporate local search procedures in order to optimize the GA results.**

This paper has proposed and implemented the integration of routing optimization using HGA with the ITU-T architecture for QoS resource control in Next Generation Networks (NGN). The implementation has been done over an IPv6 Linux testbed with OPSPv3 using the ITU-T proposed COPS-PR protocol for the policy delivery, in this case the weight setting delivery.

Keywords- QoS; PBNM; ITU-T; NGN; traffic engineering; routing optimization; OSPF; IPv6; hybrid genetic algorithm; local search

I. INTRODUCTION

The Telecommunication Standardization Sector of the International Telecommunications Union organization (ITU-T) has developed a generic end-to-end architecture for QoS resource control in Next Generation Networks (NGNs) [1]. This architecture proposes a centralized management of quality of service (QoS) through policy based network management (PBNM). For the intra-domain's policy delivery the ITU-T recommends among others the use of COPS-related protocols for the resource and admission control functions (RACF), which carries out the resources and admission control of the transport subsystem (QoS) within access and core NGNs.

One of the key factors when providing QoS is traffic engineering, given the overflows that certain links may have due to fluctuating traffic demands which can occur even in well dimensioned networks [2]. In order to tackle this problem, routing optimization aims at the optimization of networks so that more traffic can be routed in providing a possible solution to this problem. One way of achieving it is to modify the link weights and therefore the metrics. Depending on the dimensions of the topology, this weight setting problem may become a NP-hard problem [3], which can be solved through heuristics with artificial intelligence techniques. Among the different possible approaches, GAs have been shown to be appealing techniques to solve those type of NP-hard problems [4][5].

The purpose of this paper is to obtain a GNU/Linux system to optimize routing of NGNs with HGA algorithms using ITU-T specifications for QoS policy delivery. In order to optimize routing in NGNs with the aforementioned requirements we propose to use a centralized architecture, where a decision server (RACF entity) applies an offline routing protocol over the known network topology and uses an hybrid genetic algorithm (HGA) to decide the optimal weight setting of the links, which are later delivered to the nodes by the Common Open Policy Service (COPS) protocol [6] as it can be seen in Fig. 1. Thus using the centralized architecture recommended by the ITU-T. This paper extends a previous work by the authors [7] by integrating the ITU-T architecture and applying the results to a GNU/Linux testbed.

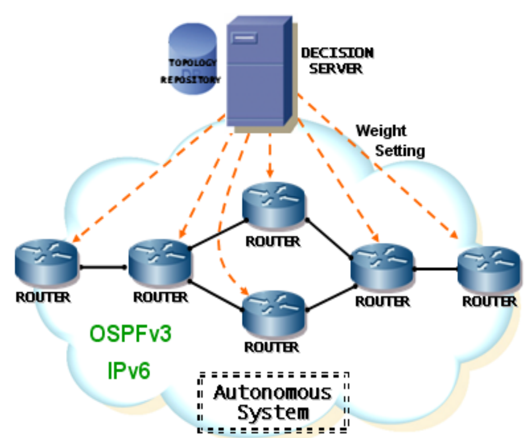

Figure 1. Network architecture for an offline routing optimization in NGNs

Therefore, the paper is structured as follows: In section II the ITU-T's RACF entity for QoS resource control management is introduced. In section III the routing fundamentals needed to understand the principles of the OSPF weight setting problem are described, in section IV the genetic algorithms (GA) for its application on routing optimization are introduced, and in section V the effects of a local search algorithm applied to the GA are illustrated. In section VI most relevant HGA algorithms are evaluated over a unique scenario which provides us with comparative data. In section VII a routing optimization proposal according to ITU-T specifications to be implemented in a GNU/Linux testbed is presented and finally, in section VIII the conclusions are provided.

II. QOS MANAGEMENT IN ITU-T'S NGNS

The generic end-to-end architecture of the ITU-T for the QoS resource control in NGNs has been developed summarizing the local efforts of different agents in their respectively field (3GPP, DSL forum, WiMAX forum, CableLabs, etc.) and the ETSI-TISPAN's generic access networks architecture.

This architecture incorporates the IP Multimedia Subsystem (IMS) architecture, developed by 3GPP, as a support of session-based services and other session initiation protocols (SIP) [8]. Therefore the QoS management architecture proposed by ITU-T is completely integrated and interoperable with IMS and can provide the management of new services and multimedia communications through diverse NGNs.

The ITU-T NGN architecture presents a stratified division between the transport plane in the lower layer and the service plane in the upper layer with a session control plane in the intermediate control sub layer (Fig. 2). The transport plane supports the transport functions and the resource and admission control functions defined in the RACF entity [1]. This entity is capable of managing the end-to-end QoS through both core and access heterogeneous networks and therefore is responsible of managing the QoS resources of every domain.

The resource and admission control functions (RACF) provide the service layer (in concrete to the SCF) with an abstract vision of the network infrastructure through a unique contact point. Thus when a service request is received from a user, the current network resource usage status is checked to see whether the requested service can be offered with a guaranteed QoS.

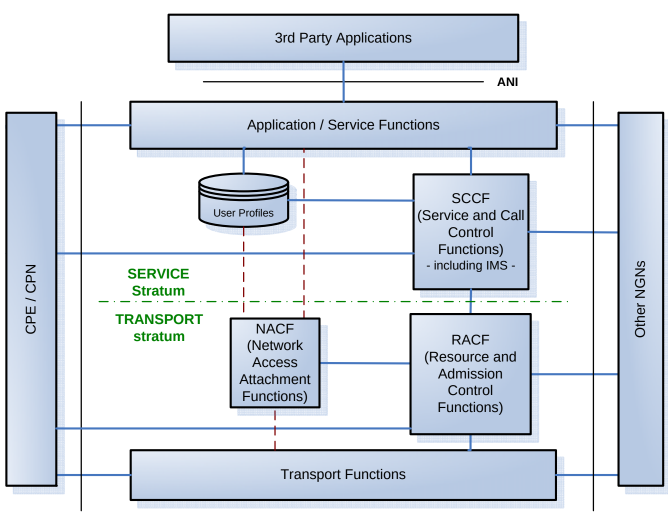

Figure 2. ITU-T NGN Framework Architecture

A. The RACF architecture

The RACF entity has adopted the policy-based network management in order to provide an efficient and centralized system to control the QoS of each domain and to provide an end-to-end QoS management tool for the end-to-end multimedia sessions which traverse domains without service stratum. This entity carries out the policy-based physical resource control, establishes the availability of these resources, decides on admission and applies the controls required to enforce the accomplishment of the policy decisions.

RACF uses reference points to manage the negotiated QoS through the session signaling and the flow control at a network level (Fig. 3). This architecture is presently under discussion and therefore some of the reference points are still to be specified. Even though different international organizations (ITU-T, ETSI-TISPAN, MSF, etc.) have proposed many alternative protocols to the network policies delivery in use in the Rw, Rc and Rn interfaces, industry has not opted for any of them.

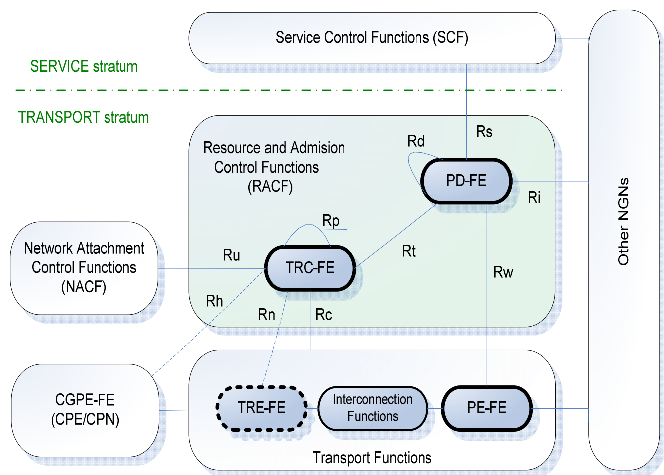

Figure 3. ITU-T RACF architecture

The Policy Decision Functional Entity (PD-FE) takes the final decision over the resource and admission control and delivers it to the corresponding Policy Enforcement Functional Entity (PE-FE) through the Rw interface. ITU-T is currently studying three alternatives for this interface, as specified in the Q.3323.x sub-series [9]: COPS-PR, H.248 and DIAMETER. The three of them are revisions of the Q.3303.x series and are still under discussion (targeted for December 2009).

The Transport Resource Control Functional Entities (TRC-FE) deals with the control of the resources which depend on transport technology. These entities are responsible for preserving and maintaining the network topology and resource database (NTRD) of each subdomain. The Rc interface is used to check the network topology and the status of network resources. The TRC-FE assigns resources to each QoS requesting flow. ITU-T has approved two alternatives for this interface, which have been specified in the Q.3324.x sub-series [10]: COPS-PR and SNMP. These specifications are revisions of the Q.3304.x series and are targeted for December 2009.

The scope and functions of the Rn and the Rh interfaces are also still under study, though it does clarify that one of the functions of the TRC-FE entity is to assign the network resources for its application and is not any of the Rc interface functions. Although ITU-T has not made any formal proposals some of the protocols which meet the requirements include SNMP and COPS-PR [11].

In this paper we are only concerned with intra-domain QoS policies, given that routing optimization will only be applied to interior gateway protocols (IGPs) and that neither the service stratum nor the user profile are required. The rest of the interfaces defined by [1] are therefore out of the scope of this paper.

Rw, Rc and Rn interfaces provide centralized management of QoS resources inside a domain and the protocol and procedures used for this task may be also be used for routing optimization. The most appropriate protocol to be used must then be determined.

B. Intra-domain control of QoS resources

Two possible scenarios are defined by the RACF entity for the QoS resource control which are based on the type of sessions established by the user. These scenarios may be in push mode or in pull mode depending on the different QoS signalling capabilities of the costumer premises equipment (CPE) that initiate the sessions and the access technology (Fig. 4).

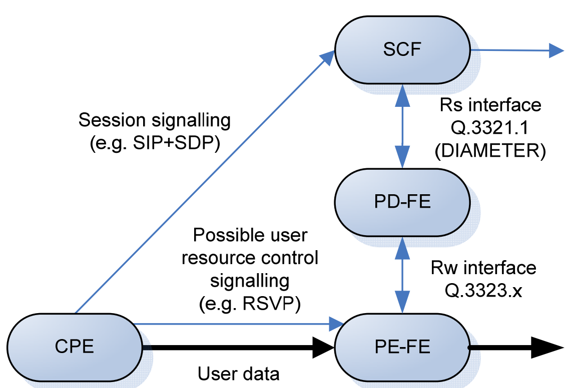

Figure 4. Push or pull operation depending on the type of user terminal

When user QoS control signalling does not exist a push mode scenario is initiated through the session signalling. The SCF are responsible for deriving the QoS needs of the requested service and sending the request to the RACF (PD-FE) for QoS authorization and reservation, sending the QoS policy to the network transport equipment (PE-FE). This mode is employed by CPE without QoS negotiating capacity (type-1 CPE) or by those with only service stratum negotiating capacity (type-2 CPE).

When the transport functions require signalling to perform a flow (e.g. RSVP o NSIS) a pull mode is initiated. In this mode it is the PE-FE which sends a QoS resource request to the PD-FE through the Rw interface, so that the RACF may take the appropriate authorization decision and replay with the final policy decision to be applied. This mode is used by CPE that explicitly request the QoS resource request through path-coupled QoS signalling (type-3 CPE).

As discussed in [12] the two protocols proposed by the ITU-T to manage the Rw interface in IP networks, DIAMETER and COPS-PR, have similar problems due to the natural client-server role (pull mode). The COPS-PR protocol works efficiently in pull mode but not so well in push mode. On the other hand, the original definition of the DIAMETER protocol also works correctly in pull mode but in this case the push mode is not contemplated. Moreover, even though most of the protocols defined in RACF interfaces and routing elements of the transport layer are still under definition, the common protocol for all of them will probably be COPS-PR. An additional factor to consider is that the COPS-PR which manages policy delivery on a native level is already supported by most IP-IP gateways (routers) in present-day networks.

C. The COPS-PR protocol

The COPS-PR protocol [11], the provisioning model, is the COPS protocol variation created to deliver QoS policies between the PD-FE and the PE-FE for the DiffServ model. Within this model, the PD-FE proactively sends the network policies to be applied to the PE-FE.

In this model the PD-FE may proactively provision the PE-FE and both have a virtual container called PIB (Policy Information Base) where the policies are stored. This PIB has a tree structure formed by PRovisioning Classes (PRCs) which contain PRovisioning Instances (PRIs) [13]. Once the PE-FE has been initiated, and whenever there are updates, the appropriate policies are sent out by the PD-FE (Fig. 5). This way the PD-FE keeps the two PIBS synchronized.

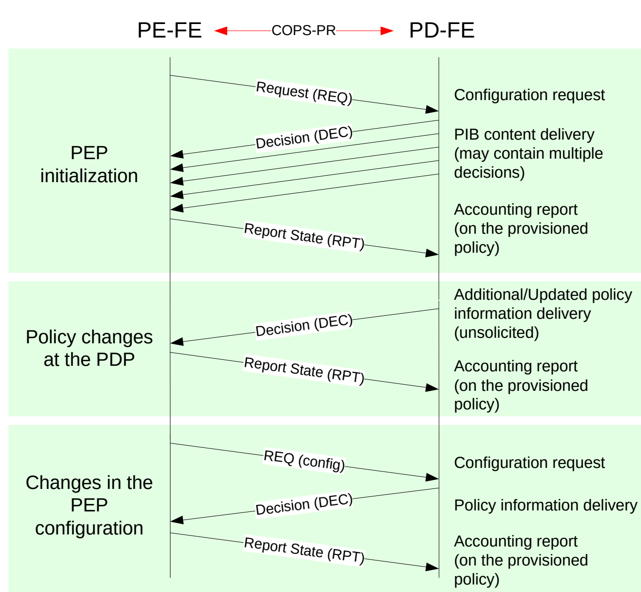

Figure 5. COPS-PR signaling between the PDP and the PEP

As stated before, when COPS-PR is applied to the RACF entity, the protocol works effectively in pull mode, but not so well in push mode, even though it does work correctly. In Fig. 6 we can see how this latter mode incorporates appended messages between the PD-FE and the PE-FE when the states of the different events which appear (detected by the PD-FE) are associated to the specific operations of the protocol.

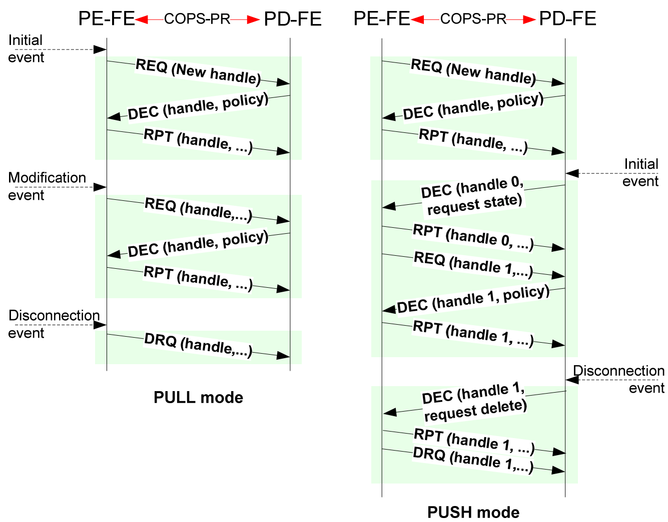

Figure 6. COPS-PR signalling in pull and push modes

In this paper, we propose the incorporation of the router link weights of the managed domain selected for routing optimization into the PIB policies with the aim of setting all the QoS parameters to the router through a single protocol, COPS-PR.

III. ROUTING OPTIMIZATION FUNDAMENTALS

In traffic engineering, a network optimal performance is generally accepted as being one where network congestion has been minimized in all the links so that all of them are equally congested. This is done from the existing resources utilization in a domain and the traffic demand matrices. One possible way of optimizing this performance is to manipulate the routing process of the packets, that is to say, by modifying the routing protocol.

The function of the routing protocols is to find the best route between an origin node and a destination node, from a minimization of the cost function. Given that the metrics provide a comparative measure to decide which path is better, we must make sure that the values of their components are properly adjusted so that they improve the global performance. Therefore to optimize routing we need to define an objective function which takes into account the link's usage in a quantifiable way and hence considers the routing cost.

The general routing problem can be described as the problem of optimizing the minimization of an objective function from a given network topology and a traffic demand matrix. Under stationary conditions the problem can be solved with linear programming (LP) and the result will be the best possible routing for all the possible flows so all the traffic will be globally optimized for all the networks in the domain.

Existing research lines tend to use the OSPF and ISIS protocols. The advantage of these protocols is that they incorporate IPv6 versions broadly used by commercials routers (RFC-5340 and RFC-5308). Moreover both protocols have versions oriented to traffic engineering with support for IPv6 (RFC-5329 and RFC-5305). These characteristics make them the most commonly-used protocols in NGN-related research. In this paper we have used OSPF, in concrete its version 3 (RFC-5340).

In order to fully understand how an effective approach to optimize overall network performance with OSPF and before considering the options available to achieve this optimization, we must first explain the how the OSPF protocol behaves.

A. *Link state routing*

The OSPF protocol is one of the most common link state routing protocols in packet networks. This protocol works with a metric based in a cost value associated to each link and applies Dijkstra's shortest path algorithm [14] to find out the shortest path to each network based on these costs. In OSPF the metric of a path to a given network is the sum of all the costs to that network.

In the case where multiple paths exist to destination with equal metrics, OSPF can balance the load with equal cost multiple path (ECMP) so that the traffic flows will be theoretically evenly split between all the paths with the same metrics. Even though it is typically impossible to guarantee an exact even split of the load, it was decided to compare OSPF with ECMP and OSPF without ECMP using an exactly even traffic distribution to achieve the simulations of this paper.

Once the initial convergence phase has finished, any change which occurs in any link, for example a weight modification, will result in only the affected link's modification being flooded. Each one of the routers will have to decide whether or not to recalculate all the information in the routing table. Therefore if the number of changes of the link weights is high, it may lead to an inefficient use of the net's resources, as well as in bandwidth as in CPU.

In this paper the inverse of the link capacity has been used as a default configuration of the link weights (or costs). This methodology was first proposed by Cisco [15] and according to [3][16] is the best way to adjust the link weights in default configurations with OSPF.

B. *The OSPF weight setting problem*

Given a known network topology and a predictable traffic demand, the OSPF weight setting problem (OSPFWS) is to find a set of weights which optimize the network performance and therefore minimize the cost function [3][17]. This problem, as stated previously, can be NP-hard depending on the dimensions of the topology. As this kind of problems can not be solved in a polynomial time, then heuristic search methods must be employed to find the most optimal solutions.

The use of local search heuristics, which apply an iterative process to solve the problem, only work for medium sized networks at the most and they do not guarantee the best possible solution [18]. The most commonly-used algorithm is the one proposed by Fortz and Thorup [16], which proposes minimizing a function that summarizes all the link weights so that it optimizes the global performance of the domain. This proposed cost function is convex, incremental, lineal, continuous and piece-wise, which assigns low costs to the infrequently used links and high costs to the overloaded links. If the problem needs to be solved for medium-big sized networks it is necessary to use artificial intelligence, given that using a local search heuristic is not viable in computing time terms.

IV. GENETIC ALGORITHM HEURISTICS FOR ROUTING OPTIMIZATION

Genetic algorithms [4][5] are methods for search, optimization and machine learning which are inspired by natural principles and biology. Differently from other optimization methods, GAs do not assume any structure or underlying distribution of the objective function and employ random, local operators to evolve a population of potential solutions. Since GAs have demonstrated to be able to solve complex problems that previously eluded solution [19], we have chosen to adopt this optimization model in our design. In the following, we first present the basic mechanics of GAs and then explain different GA implementation to solve the routing optimization problem.

A. Mechanics of genetic algorithms

GAs evolve a *population* of individuals, where each of them represents a potential solution to the problem. Analogous to genetics, individuals are represented by *chromosomes*, which encode the decision variables of the optimization problem with a finite-length string. Each of the atomic parts of the chromosome is referred to as *genes*, and the values that the gene can take are addressed as *alleles*. To implement the principles of natural selection and competition among candidate solutions, GAs incorporate an *evaluation function* that gives a certain value of *fitness* to each individual, which indicates the quality of the given individual.

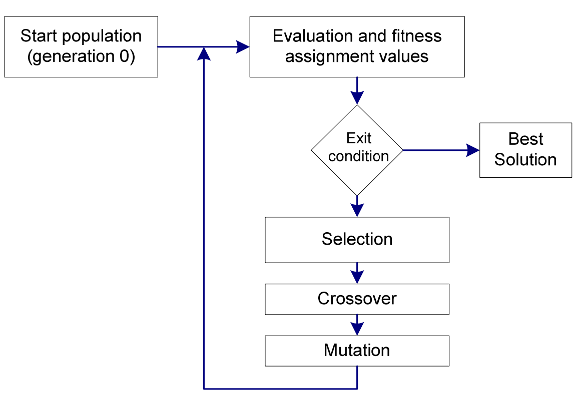

Figure 7. Flow chart of a Genetic Algorithm

Then, this population of individuals, which is usually initialized randomly, is evolved by a continuous process of *selection*, *crossover, mutation*, and *replacement* of individuals. That is, firstly the selection operator chooses the fittest individuals in the population, simulating the survival-of-the-fittest mechanism. Then, the crossover operator takes two or more of the selected individuals and recombines their genetic information in order to generate new, possibly better offspring. Afterwards, mutation introduces random errors on the transference of genetic information from parents to children, and finally, the offspring population replaces the original one. This process is repeated until a stop criterion is met; usually, the process is run during a prefixed number of iterations. Fig. 7 schematically illustrates this process.

The synergy of all these operators pressures toward the evolution and selection of the best solutions, which are recombined yielding new promising offspring. In [19], Goldberg emphasized the idea that, while selection, crossover, and mutation can be shown to be ineffective when applied individually, they might produce a useful result when working together. This was explained with the fundamental intuition of GAs, which supports the following two hypotheses. The first hypothesis is that the combination of the selection and crossover operators introduces a process of innovation or cross-fertilizing by generating new solutions from the fittest individuals in the population. As a consequence, new individuals are expected to be different from and better adapted than their parents. The second hypothesis is that the combination of selection and mutation represents a process of continuous improvement or local search. Thence, this process searches around the best solutions in the population with the aim of finding better solutions that are close to the parents.

B. Design of a genetic algorithm for routing optimization

With the basic mechanisms of GAs in mind, now we are in position to proceed with the description of how GAs have been applied to the routing optimization problem. In what follows, we present the typical representation employed by several authors, and discuss which types of genetic operators have been used in different approaches [20][21][22][23][24].

In order to solve the OSPFWS problem, the solution must contain the weights of each link of the network. Therefore, the individual is typically represented as a vector that contains all the link weights of the domain, which range in [1, w_{max}]. Thus, each individual provides a complete solution to the problem. This population is typically initialized with the default weight setting with the inverse capacity procedure as proposed by Cisco [15]. On the other hand, several evaluation functions that provide a measure of the link performance of the domain, showing the most overloaded or the global average performance, have been used by different authors. In the following section three of these functions are explained in more detail.

Different genetic operators have been used in different GA implementations for this problem so far. Two selection schemes were employed in the approaches studied in this paper: rank selection and proportionate or roulette wheel selection. Rank selection ranks the individuals of the population according to their fitness, and those with better ranking are selected to be in the next generation. An especial case of rank selection is tournament selection, which uses a set of *s* randomly selected individuals for ranking instead of considering the whole population. On the other hand, proportionate selection gives each individual a selection probability that is proportional to its fitness with respect to the fitness of the other individuals in the population. Optionally in this phase a technique named elitism can be used, which consists in always passing at least one copy of the best chromosome to the next population, so the best individuals are not lost due to the effect of the genetic operators.

Then, crossover and mutation reproduce the parent population as follows. Crossover is applied to each pair of parents with a certain probability. If applied, crossover randomly generates a cut point and uses this cut point to shuffle the genetic information of the parents. Therefore, crossover creates two new individuals that mix the genetic information of the parents. If crossover is not applied, the offspring are exact copies of the parent. Thereafter, the offspring undergo mutation. That is to say, for each gene, a random number is generated and, if it is lower than the probability of applying mutation, the gene is mutated by assigning a new randomly selected value in the interval [1, w_{max}]. After mutation, the new population replaces the original one.

A representative example of a GA applied to routing optimization is provided by Ericsson et al. [20]. In this case the GA is based on the idea proposed by the heuristic search in [3][17] and it applies the same cost function. The representation of the population's individuals is formed by the set of all the link weights in a vector. The population initialization is randomly generated and the selection method is the rank selection where it divides the population in three sets of α=20% (elitism), β=70% (crossover) and γ=10% (discarding), in respect the total population size between 50 and 500 individuals. The crossover probability is 70%, mutation probability is 1% and the number of iterations is variable between 500 and 700.

Throughout this section, we have explained the process organization of GAs, have intuitively discussed how and why they work, and have shown how GAs have been applied to the routing optimization problem. In the next section, we take these ideas and explain how GAs can be enhanced by incorporating a new local search procedure that enhances the original local search mechanism of GAs – that is, mutation – in order to converge quicker to the objective.

V. LOCAL SEARCH WITH GA FOR ROUTING OPTIMIZATION

The hybrid genetic algorithms (HGA) [21], or memetic algorithms, are distinguished from the GA because they append a local search heuristic applied during the evolutionary cycle, as can be seen in a typical HGA flowchart in Fig. 8. The objective of this local search procedure is to improve the effectiveness and efficiency of a GA when converging to an optimal solution of the problem. In this section we provide an inedited theoretical comparative analysis of the main three HGA proposals.

The hybrid genetic algorithms obtain better results when optimizing the global performance of the domain with OSPF routing process rather than the simple GA [22][23][24]. Some benefits that the local search addition provides are acceleration in the optimization process (in computational time) and improvement in the quality of the solutions, which are more optimized, that is to say better, as demonstrated in [23] and [24]. However, a disadvantage is the potential loss of the global maximum, getting stuck in a local maximum, as happens when there is an abuse of the genetic operators.

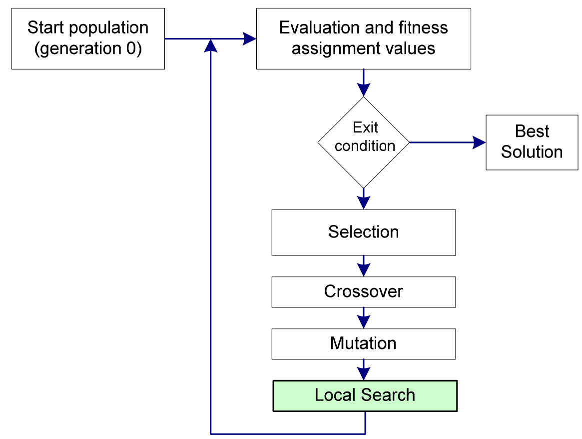

Figure 8. Flow chart of a Hybrid Genetic Algorithm

A theoretical analysis of the HGA algorithms proposed in [22], [23] and [24] has been carried out. Table I presents a summary of the most representative genetic parameters of the three proposals. All of them have used the same values in the crossover parameter and mutation, but they differ in the selection method, weight representation and overcoat, where to apply the local search procedure and the fitness function.

TABLE I. PARAMETERS OF THE HYBRID GENETIC ALGORITHMS

	Mulyana & Killat [22]	**Buriol et al. [23]**	**Riedl & Schupke [24]**
Population Size	50	50	20
Chromosome Representation	Set of all domain's link weights in a vector		
Weight Representation	[1, 99]	[1, 20]	[1, 20]
Number of Iterations	200	200	200
Selection Method	Rank Selection: α=20%, β=70%, γ=10%	Rank Selection: α=25%, β=70%, γ=5%	Roulette Wheel Selection
Crossover Probability (P_c)	0.7	0.7	0.7
Mutation Probability (P_m)	0.01	0.01	0.01
Heuristic Search	Best individual	Individual generated from the crossover	All individual

The fitness function used by Buriol et al. in [23] is the same as the one used by Resende et al. in [20], in the first application of a GA to the optimization problem and has been introduced in the previous section. Both of them use the convex cost function proposed by Fortz and Thorup [16].

$$\min \phi = \sum_{a \in A} \phi_a \tag{1}$$

subject to

$$\sum_{u:(u,v)\in A} f^{(st)}_{(u,v)} - \sum_{u:(u,v)\in A} f^{(st)}_{(v,u)} =$$
$$= \begin{cases} -d_{st} & \text{if } v = s, \\ d_{st} & \text{if } v = t, \quad v, s, t \in N, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$\ell_a = \sum_{(s,t)\in NxN} f^{(st)}_a, a \in A, \tag{3}$$

$$\phi_a \geq \ell_a, a \in A, \tag{4}$$
$$\phi_a \geq 3\ell_a - 2/3c_a, a \in A \tag{5}$$
$$\phi_a \geq 10\ell_a - 16/3c_a, a \in A \tag{6}$$
$$\phi_a \geq 70\ell_a - 178/3c_a, a \in A \tag{7}$$
$$\phi_a \geq 500\ell_a - 1468/3c_a, a \in A \tag{8}$$
$$\phi_a \geq 5000\ell_a - 19468/3c_a, a \in A \tag{9}$$
$$f^{(st)}_a \geq 0, a \in A; s, t \in N \tag{10}$$

Later, Mulyana and Killat [22] tackled the General Routing Problem by minimizing the following fitness function, which takes into account a weighted addition of the global average and the maximum link utilization.

$$min \left[(a_t \cdot t) + \frac{1}{|E|} \sum_{ij} \sum_{uv} \frac{\ell^{uv}_{ij}}{c_{ij}} \right]$$
$$\forall (i,j) \in E, \forall (u,v) \in V \times V \tag{11}$$

$$\delta_{un} f_{uv} + \sum_{m\in V} \ell^{uv}_{mn} = \delta_{nv} f_{uv} + \sum_{m \in V} \ell^{uv}_{nm}$$
$$\forall (u,v) \in V \times V, \forall n, m \in V \tag{12}$$

$$\sum_{uv} \frac{\ell^{uv}_{ij}}{c_{ij}} \leq t, \forall (i,j) \in E$$
$$\ell^{uv}_{ij} \geq 0, \forall (i,j) \in E, \forall (u,v) \in V \times V \tag{13}$$

Finally, the fitness function employed by Riedl and Schupke in [24], only considers minimizing link maximum utilization with the application of an exponential scalability factor. This way, routing solutions with smaller maximum link utilization receive higher fitness values and a greater chance to be reproduced in the new generation.

$$fitness = \left(\frac{1}{\rho_{max}} \right)^p, p > 0. \tag{14}$$

subject to:

$$\rho_{max} \geq \rho_{ij} \quad \forall (i,j) \in \varepsilon \tag{15}$$
$$\rho_{ij} = \sum_{u \in V} \frac{f_{ij,u}}{C_{ij}} \quad \forall (i,j) \in \varepsilon \tag{16}$$

VI. EVALUATION OF THE HGAS

The aim of this section is to choose one of the HGAs proposed in the previous section and consequently implement it in a real testbed, providing a comparative analysis of the main three HGA proposals. The main problem when comparing the different algorithm's proposals of the authors is the fact that each one uses its own network topologies and traffic matrices. As the objective is to evaluate comparing the algorithms it is necessary to determine a common topology and traffic demand matrix to apply and test them. When this comparative was first proposed by the authors in [7] there was a physical limitation of twelve routers creating a restriction regarding the topology of the test. Therefore the network topology N11 used in [24] has been selected. This network topology, which has 11 nodes and 48 unidirectional links, can be seen in Fig. 9.

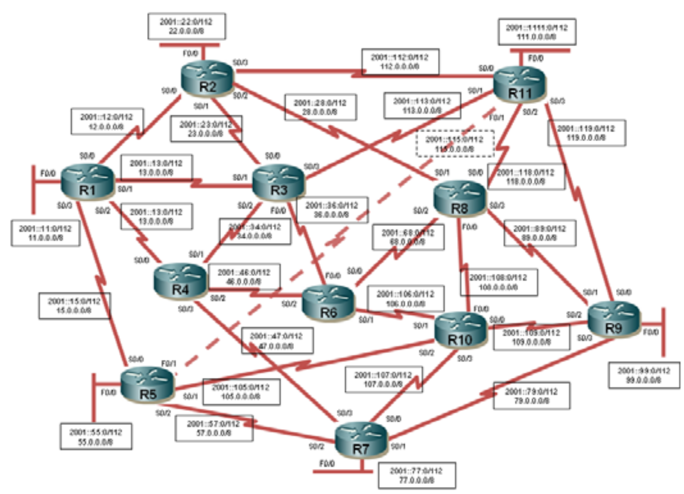

Figure 9. Network topology N11 used to evaluate the routing optimization

In order to evaluate the proposals comparisons between the different approaches have been made: the default inverse capacity metric (InvCap) [15], the local search algorithm proposed by [17], the GA proposed by [20] and the three HGAs proposed by [22], [23] and [24]. The parameters proposed by the authors of the HGA algorithms have been used in this evaluation. Algorithms with a maximum number of iterations of 250 and a maximum percentage of total weight change of 30% from initial configuration were used in order to avoid excessive flooding which would have consequently led to excessive routing re-calculations. Six traffic demand matrices with increasing traffic have been generated.

The graphs show how, in general, the HGA algorithms improve local search and simple GA algorithms' results. The two graphs, OSPF without ECMP (Fig.10) and OSPF with ECMP (Fig.11), show how the HGA algorithm proposed by Buriol et al. [23] is the one that most effectively minimizes the maximum usage of the most congested link. Overcoat in the worst traffic cases when under default conditions (InvCap) there is link overloaded, even though there exist minimum differences with respect to the other two HGA proposals.

On the other hand the HGA algorithm proposed by Mulyana and Killat [22] is the one which best minimizes, albeit marginally, the average usage of the links in all cases, as can be seen in the graphs of OSPF without ECMP (Fig.12) and OSPF with ECMP (Fig. 13). In the case of average usage of the links, even though the HGA algorithms always provide better results, the deviation with the others are minimal in percentage

According to results minimal variations between the three HGA proposals exist, both in the average usage as well as in the maximum usage. It was decided to use the algorithm proposed by Buriol et al. [23] because it presents slightly better results in this latter aspect.

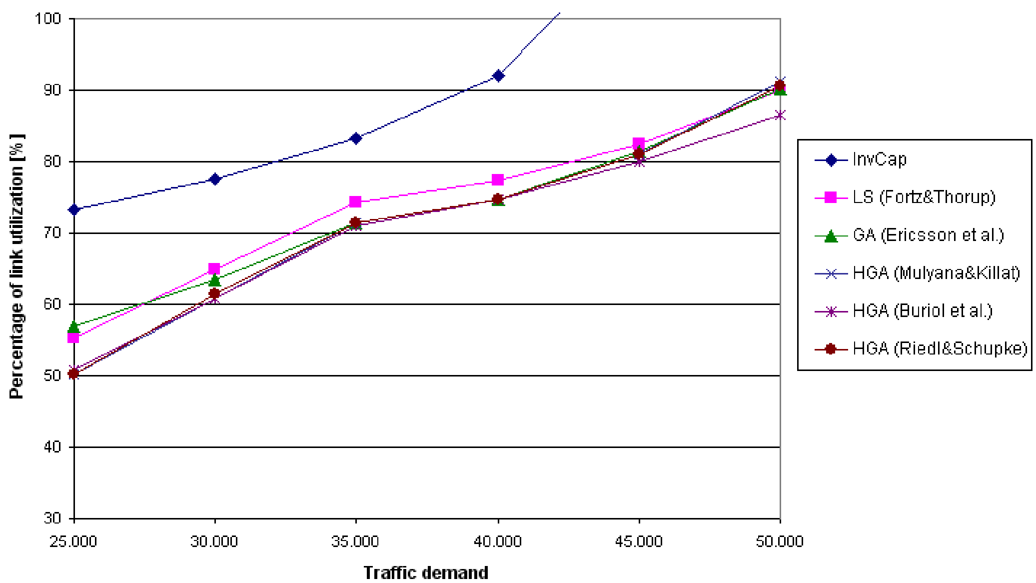

Figure 10. Maximum link utilization without ECMP

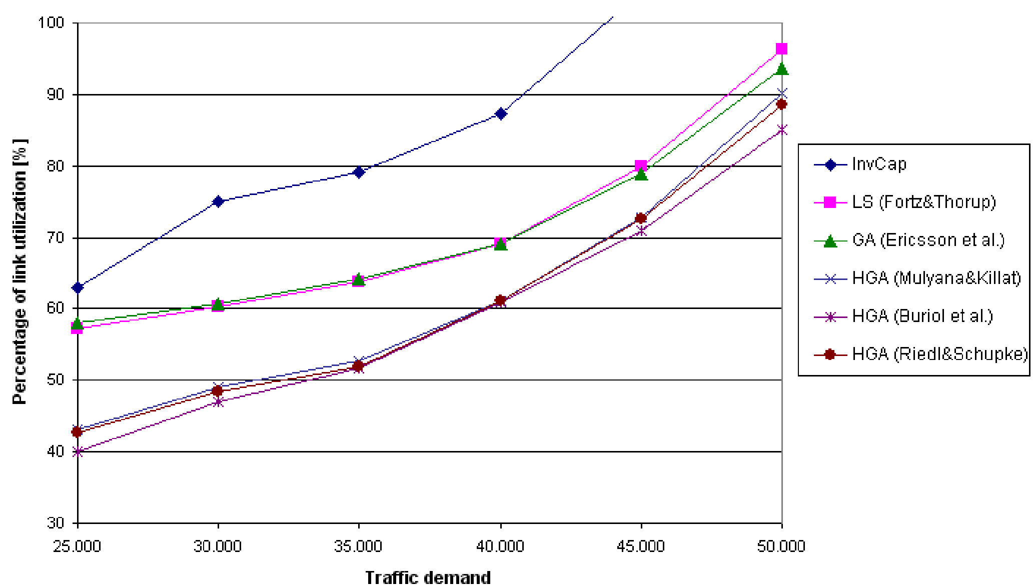

Figure 11. Maximum link utilization with ECMP

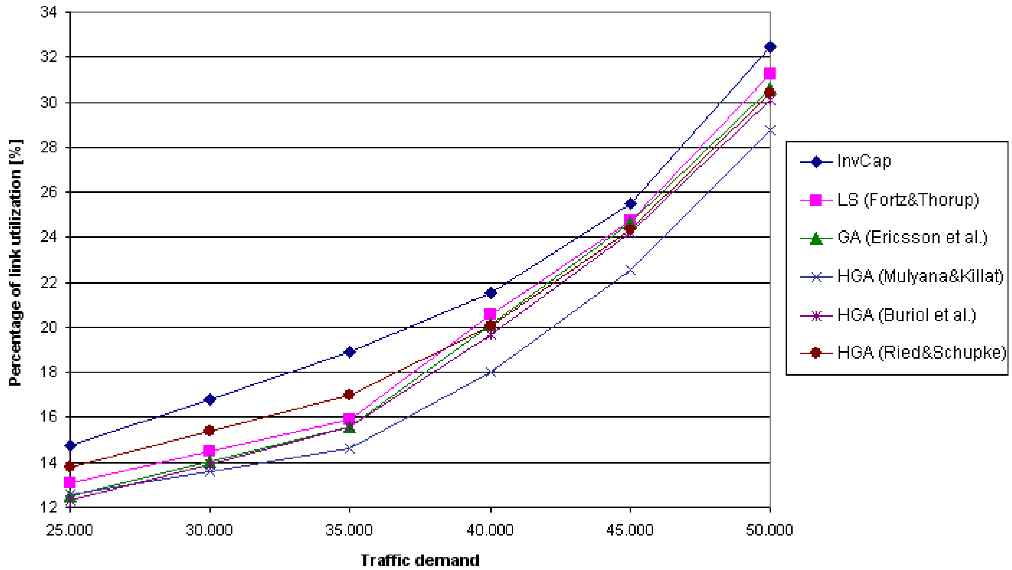

Figure 12. Average link utilization without ECMP

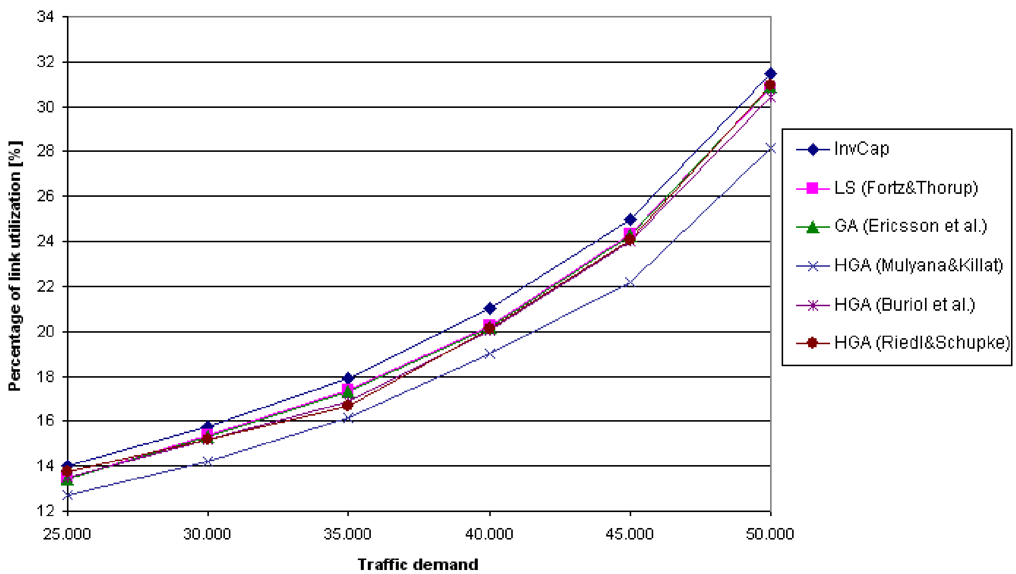

Figure 13. Average link utilization with ECMP

VII. TESTBED IMPLEMENTATION

This section presents a real testbed implementation where it has been upgraded the routing optimization presented in [7] to meet the ITU-T requirements. Thus, in order to deliver the link weights to the routers over a NGN testbed we have used the COPS-PR protocol

The authors presented in [7] a successful implementation of routing optimization in an IPv6 domain with eleven commercial 262x Cisco routers and with a centralized GNU/Linux device. The task of this centralized device was computing the link weights through a HGA algorithm and sending them to domain routers, where they were configured automatically (Fig. 1 shows the architecture). In the aforementioned paper the sending of the optimized link weights was carried out via the SSH and the routing protocol used was OSPFv3.

In this paper we have integrated the application developed in [7] into the NGN testbed presented in [12], where we implemented an end-to-end QoS management signaling proposal for the ITU-T NGN architecture. In this latter paper the COPS-PR protocol was used to manage the intra-domain policies and resources with the GNU/Linux routers (Rw, Rc and Rn interfaces) and the COPS-SLS protocol was used to dynamically negotiate the inter-domain policies among the centralized devices of each domain (Ri interface). These devices are formally the RACF entity and have been named as QoS Brokers (QoSBv6).

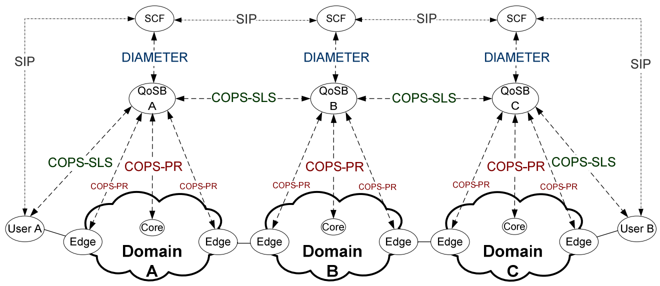

Figure 14. Proposed architecture in [12] for end-to-end signalling

We have used an application in the QoSBv6 to compute the offline routing from the domain's topology and the traffic matrix with the OSPFv3 protocol. This application was developed in [7]. The default link weights of the GNU/Linux router interfaces are obtained by using the link's inverse capacity [15] and applying the HGA algorithm proposed by Buriol et al. [23] these link weights can be modified, to obtain the new "genetically" optimized weights. The sending of these new OSPF weights to the GNU/Linux router interfaces is carried out via the COPS-PR protocol.

This way, whenever the administrator decides to optimize routing, which is relatively infrequent, the application applies the HGA to the current topology and sends the new weights to the system Database, which will be copied into the COPS-PR PIB and, later, delivered to the routers with COPS-PR.

A. System Database

The database stores the domain policies such as service level agreements (SLA) with the client traffic demand matrix of the client, the network topology (nodes, link's capacities and costs) and the addresses of the QoSBv6s of adjacent domains. The client traffic demand matrix is a maximum traffic configuration, required in order to assure client's contract traffic.

PostgreSQL v.7.4.6. is used because its JAVA connector supports IPv6 connections. The SLS policies and the link weights are transferred to the PIB for their distribution through COPS-PR to the routers or Policy Enforcement Physical Entity (PE-PE). Therefore, the relation of routers of the domain permits the installation of the network policies to the routers.

B. COPS-PR module and the PIB

The protocol used for the intra-domain communication is COPS-PR. Besides the support of IPv6 protocol, we implement the whole protocol, including the keep-alive function, the synchronization function for the PDP-PEP disconnection case and the PEP-redirect function in the case of failure. This latter function also supports PDP redirection with IPv6 addresses. Support for the PIB defined by the DiffServ WG [25] has also been included in addition to the previously determined RAP WG PIB definition [26], which always supports IPv6. Our PIB completely fulfils them.

We have modified the COPS-PR Policy Information Base (PIB) so the router link weights are sent as a piece of the policies. COPS-PR allows unsolicited sending of policies from the PD-FE to the PE-FE and, moreover, it only allows sending parts of the PIB for efficiency improvements. Therefore whenever it is necessary to change the link weights, only a small part of the PIB is sent through COPS-PR, which will be only the modified links weights.

C. The GNU/Linux router

The IPv6 protocol is currently supported by the main software and hardware components and is present in the main worldwide networks. Furthermore, as a result of the work carried out by the USAGI project [27], which has merged its work into the official Linux kernel, IPv6 stack in Linux OS is now fully compliant with advanced IPv6 conformance and interoperability tests. Therefore NGN testbeds using the IPv6 protocol with Linux OS can be implemented with performance guarantees.

The PE-PE module installed in every router fully supports COPS-PR over IPv6 and is responsible for the configuration of the policies in the PE-PE's PIB and the link weights in a computer running GNU/Linux as a router. The GNU/Linux kernel has been configured for IPv6 support with all the QoS functionalities available to be used in IPv4 as well as IPv6.

The load balancing in the routers is automatically activated if the routing table has multiple paths to a destination. In this testbed per-destination load balancing has been used, where the router distributes the packets based on the destination address. Another option could have been per-packet load balancing which guarantees equal load across all links but there is the possibility that the packets may arrive out of order at the destination if differential delay exists within the network and it is a processor intensive task which may impact the overall forwarding performance. We should underline the fact that even though per-destination load balancing is an improvement over per-packet, it is still not very good because if substantially more packets are sent to one destination than to another, the overall bandwidth utilization will be uneven.

D. From theory to real world

Thus far, we have proposed a traffic engineering method to be applied under ITU-T NGN specifications and we have proven its viability by using COPS-PR for the router link weights delivery. We have successfully optimized weights of a single GNU/Linux router applying the offline algorithm successfully proven in simulations and using the NGN testbed successfully deployed in [12]. Despite this headway progress, in this paper we have not optimized and evaluated a complete system with this new proposal even though we are currently working on it.

Nevertheless it must be remarked the great difficulty of the hand-on testbed versus simulation implementations. Moreover the fact of haven built our testbed over physical machines, it provides an extra difficulty over the emulated solution. The preliminary results show the viability of our proposal in a real environment and using open systems. Therefore, this paper represents the baseline for future work detailed in the next section.

VIII. CONCLUSIONS AND FURTHER WORK

The ITU-T MS/NGN architecture must provide support for the QoS of the multimedia sessions. This support includes the QoS negotiation, admission and resource control for different end-to-end QoS models. Therefore, it seems reasonable to integrate traffic engineering for intra-domain optimization into this architecture. Assuming this hypothesis, this paper has presented the implementation of a traffic engineering proposal in ITU-T NGN environment by means of routing optimization. This optimization has been done through the application a HGA algorithm to offline routing with the aim of modifying the OSPF link weights and therefore minimizing the maximum utilization of the domain's links.

A comparative study of the various proposals to solve the OSPFWS problem has been carried out and it has been demonstrated that HGA algorithms provide good solutions to the complex problem, better than those provided by GA algorithms. Among those, the one which provided slightly better results was the Buriol et al. proposal and which was hence selected to be implemented in the testbed.

The testbed implementation has been done with GNU/Linux routers and with a centralized QoSBv6 device. This device had the task of computing the link weights through the elected HGA algorithm and sending them to the domain's routers, where they are configured automatically. OSPFv3 is the routing protocol which decides the routing inside the autonomous system implemented with IPv6 and computed with the optimized weights. The sending of the optimized link weights has been carried out with the COPS-PR protocol, proposed by the IMS/NGN architecture to manage domain's internal policies with PBNM.

We are currently working on extending the work presented in this paper to optimize and evaluate a complete system with this new proposal. On the other hand, given that one of the functions of the QoSBv6 in [12] is to manage QoS DiffServ model policies inside a domain, in [28] a description of how routing optimization can be applied to the DiffServ model is provided. This testing will be also undertaken in further work.

ACKNOWLEDGMENT

We would like to thank "Enginyeria i Arquitectura La Salle" (University Ramon Llull) for their encouragement and assistance, especially C. Duz and D. Cutiller for his support throughout the development of this paper.

REFERENCES

[1] ITU-T Rec. Y.2111, "Resource and Admission Control Functions in NGN (version 2)". November 2008.

[2] X. Xiao, L. Ni, "Internet QoS: A Big Picture", IEEE Network, pp. 8-18, March 1999.

[3] B. Fortz, M. Thorup, "Internet traffic engineering by optimizing OSPF weights", Proceedings of IEEE INFOCOM 2000, pp. 519-528, March 2000.

[4] J. H. Holland,"Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artiçial Intelligence", MIT Press/ Bradford Books edition, 1992.

[5] D.E. Goldberg, "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley, Massachusetts, 1989.

[6] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", IETF RFC 2748, January 2000.

[7] A. Vallejo, et al., "Implementation of Traffic Engineering in NGNs Using Hybrid Genetic Algorithms", Proceedings of ICSNC 2008, pp.262-267, October 2008.

[8] ITU-T Rec. Y.2021 "IP Multimedia Subsystem for NGN". September 2006.

[9] ITU-T Draft Recommendations Q.3323.x: "Protocol at the interface between Policy Decision Physical Entity (PD-PE) and Policy Enforcement Physical Entity PE-PE (Rw interface) version 2", 2009.

[10] ITU-T Draft Recommendations Q.3324.x: "Protocol at the interface between Transport Resource Control Physical Entity (TRC-PE) and Transport Phisical Entity (T-PE) (Rc interface) version 2", 2009.

[11] K. Chan, et al., "COPS Usage for Policy Provisioning (COPS-PR)", IETF RFC 3084, March 2001.

[12] A. Vallejo, A. Zaballos, X. Canaleta and J. Dalmau, "End-to-end QoS management proposal for the ITU-T IMS/NGN architecture", Procedings of SoftCOM 2008, pp 147-151, September 2008.

[13] K. McCloghrie, et al., "Structure of Policy Provisioning Information (SPPI)", IETF RFC 3159, August 2001.

[14] E. Dijkstra, "A note on two problems in connection of graphs", Numerical mathematics, Vol. 1, pp. 269-271, 1959.

[15] Cisco Systems, "Configuring OSPF", Cisco Systems, Inc., San Jose, USA, August, 2006.

[16] B. Fortz, J. Rexford, M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols", IEEE Communications Magazine, Vol. 40, N. 10, pp. 118-124, May 2002.

[17] B. Fortz, M. Thorup, "Increasing Internet capacity using local search", Computacional optimization and applications, Vol. 29, N. 1, pp. 13-48, October 2004.

[18] M. Söderqvist, "Search Heuristics for Load Balancing in IP Networks", SICS. Technical Report T2005:04, March 2005.

[19] D.E. Goldberg, "The Design of Innovation: Lessons from and for Competent Genetic Algorithms", Kluwer Academic Publishers, 2002.

[20] M. Ericsson, M.G.C. Resende, P.M. Pardalos, "A Genetic Algorithm For The Weight Setting Problem in OSPF Routing", Journal of Combinatorial Optimization, Vol. 6, N. 3, pp. 299-333 , September 2002.

[21] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Material Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report 826, 1989.

[22] E. Mulyana, U. Killat, "A Hybrid Genetic Algorithm Approach for OSPF Weight Setting Problem", Proceedings of the 2nd Polish-German Teletraffic Symposium PGTS 2002, September 2002.

[23] L.S. Buriol, M.G.C. Resende, C.C. Ribeiro, M. Thorup, "A hybrid genetic algorithm for the weight setting problem in OSPF/ISiS routing", Networks, Vol. 46, N. 1, pp. 36-56, 2005.

[24] A. Riedl, D. A. Schupke, "Routing optimization in IP networks utilizing additive and concave metrics", IEEE/ACM Transactions on Networking, Vol. 15, N. 5, pp. 1136-1148, 2007.

[25] K. Chan, R. Sahita, S. Hahn, K. McCloghrie, "Differentiated Services Quality of Service Policy Information Base", IETF RFC 3317, March 2003.

[26] R. Sahita, Ed., S. Hahn, K. Chan, K. McCloghrie, "Framework Policy Information Base", IETF RFC 3318, March 2003.

[27] University of Tokyo, USAGI (UniverSAl playGround for Ipv6) Project, December 2008, [Online]. Available: http://www.linux-ipv6.org.

[28] B. Fortz, M. Thorup, "Optimizing OSPF/ISIS weights in a changing world", IEEE Journal on Selected Areas in Communications, Vol. 20, N. 4, pp. 756-767, May 2002.