Neural Decision Making for Decentralized Pricing-based Call Admission Control

F. Davoli, *Senior Member*, IEEE, M. Marchese, *Senior Member*, IEEE, M. Mongelli, *Student Member*, IEEE
DIST – Department of Communication, Computer and System Sciences
University of Genoa,
Via Opera Pia 13, 16145, Genoa, Italy
{ franco, dama, mopa }@dist.unige.it

***Abstract* — In this paper, a novel *Call Admission Control* (CAC) problem is investigated in relation to the pricing structure of a telecommunication network, in which both *Guaranteed Performance* (GP) and *Best Effort* (BE) services are offered. The user's sensitivity to the prices is described through utility functions. An original decision making process is studied to decentralize the proposed CAC mechanism. To this aim, a neural approximation technique is investigated to exploit different *Decision Makers*, distributed in the network and performing the CAC decisions. Simulation results show how sub-optimal CAC decisions are obtained in a decentralized fashion and with a small on-line computational effort.**

***Key words* — Utility-based Pricing, Call Admission Control, Decentralized Control, Neural Networks.**

I. Introduction

Optimization techniques for telecommunication networks usually lead to the implementation of control algorithms requiring a centralized management of the network. Since the origin of the first computer networks, decentralizing the network management, together with obtaining optimized performance, has revealed to be a hot topic of research. Even today, when telecommunication networks are still evolving towards *Quality of Service* (QoS) architectures, this area of research gives rise to different control methodologies (see [1] for an overview concerning this topic). Particularly in DiffServ and MPLS environments, where the so-called *border* (or *edge*) routers are required to manage a broad range of functionalities (classification, admission control, routing balancing, bandwidth reservation, signaling management), decentralized control is an attractive instrument to avoid a heavy centralization of the network management.

In this paper, we firstly examine a novel pricing-based *Call Admission Control* (CAC) mechanism, suitable for networks offering both *Best Effort* (BE) and *Guaranteed Performance* (GP) services. Then, we address the decentralization of such CAC by emphasizing both the simplicity of the proposed approach and the related small on-line computational effort.

The remainder of the paper is organized as follows. In the next Section we summarize the pricing-based CAC problem we deal with. In Section III, we detail our approach to avoid centralized management through a distributed neural-based decision making. In Section IV, simulation results are presented to validate the decentralized CAC and, in Section V, we finally conclude by summarizing the obtained results and emphasizing directions for future research.

II. The Pricing-based CAC Problem

Several works exploit users' satisfaction and pricing sensitivity through the concept of *utility function* [2]. It is possible to define the utility $U_r(x_r)$ to measure the benefit received by the user r when the rate transmission is x_r. In the context of pricing, it is useful to think of it as the amount of money the user *r* is willing to pay for a certain x_r.

Let a telecommunication network be composed by a set J of unidirectional links. Link j has capacity c_j; $J(r)$ is the subset of J containing the links traversed by the user r; $R(j)$ is the subset of users traversing link j. Let $\boldsymbol{A} = \{A_{jr}, j \in J\}$ be the matrix that defines the resources assigned to the users ($A_{jr} = 1$ if link j is used by user r, $A_{jr} = 0$, otherwise). In such a context, each user accessing the network maximizes its utility with respect to the assigned price p^r, i.e., its bandwidth demand x_r^o is ruled by (1):

$$x_r^o = \arg \max_{x_r \in [m_r, M_r]} (U_r(x_r) - x_r \cdot p_r) \tag{1}$$

For instance, p_r may be in terms of [€/Mbps per minute], x_r in [Mbps] and $x_r \cdot p_r$ in [€/minute]. Considering BE services, we now briefly recall the results of [3, 4] to underline that a decentralized implementation of a congestion-dependent pricing is available to achieve the maximization of the so-called *network social welfare* defined in (2):

$$\boldsymbol{x}^o = \arg \max_{x_r \in [m_r, M_r]} \sum_r U_r(x_r); \ \boldsymbol{A}^{BE} \cdot \boldsymbol{x} \le \boldsymbol{c}, \ \boldsymbol{x} \ge \boldsymbol{0} \tag{2}$$

when all the users react to prices as outlined in (1). We denote by $\boldsymbol{x}$ and $\boldsymbol{c}$ the aggregate vectors of the BE users' rates and of the link capacities, respectively. In brief, the key idea is to exploit the Lagrangian decomposition of (2), thus giving rise to a distributed algorithm of the form:

$$x_r(p_r(t)) = [U_r'^{-1}(p_r(t))]_{m_r}^{M_r}; \ p_r = \sum_{j \in J(r)} p^j \tag{3}$$

$$p^j(t+1) = p^j(t) - \eta \cdot [c_j - \sum_{r \in R(j)} x_r(p_r(t))] \quad (4)$$

Eq. (3) represents the solution of (1), $[z]_a^b = \min\{\max\{z,a\},b\}$ and $U_r^{'-1}$ denotes the inverse of the utility function derivative. At each iteration, the user r individually solves (1) through (3) and sets the rate on its path $J(r)$ to $x_r(p_r(t))$. Each link $j \in J(r)$ updates its price p^j according to (4), and communicates the new prices to the user r, whose transmission rate must be changed according to (3). Then, the cycle starts again. Such pricing mechanism (integrated within the flow control) achieves an ideal situation, in which all the users act individually by pursuing their own benefits, but, at the same time, it guarantees the maximization of the network welfare (2).

Real time traffic complicates the situation. Since it requires QoS constraints, it gives rise to GP applications whose pricing structure involves the corresponding *effective bandwidth* of the services [2]. Taking (1) as reference to depict the user's sensitivity to the pricing structure, we now investigates the presence of both BE and GP users multiplexed together in a telecommunication network.

Similarly as said above for BE traffic, a GP user s (i.e., a user that requires specific QoS guarantees) is defined as a connection established on a specific path, consisting of a non-empty subset of J, $J_{GP}(s)$; R_{GP} is the set of active GP users. We indicate with $\boldsymbol{A}^{GP} = \{A_{js}^{GP}, j \in J, s \in R_{GP}\}$ the matrix that defines the resources assigned to GP users and with $R_{GP}(j)$ the subset of GP users that use link j. Let y_s be the equivalent bandwidth required by the user s and $U_s(y_s)$ the utility function of such user. The only difference with the BE users stems from the fact that a GP user does not accept a variable bandwidth allocation. It requires a bandwidth pipe of capacity y_s and its *willingness to pay*, v_s, for such y_s is obtained from (1) as $v_s = U_s^{'}(y_s)$ (by replacing x_r with y_s and p_r with v_s).

Moreover, a GP user enters the network if there is enough bandwidth availability on the assigned routing path and only if the price imposed by the *Service Provider* (SP) is equal to or less than its willingness to pay v_s, $s \in R_{GP}$.

Let now R_{BE} be the set of active BE users. When the BE traffic is regulated by the aforementioned pricing-based flow control mechanism (3) and (4), before a new GP user $\tilde{s}$ enters the network, the rates x_r and the prices p_r of the current BE users $r \in R_{BE}$ have reached the stationary optimal values x_r^o and p_r^o by solving (2) through (3)-(4) before the arrival (or the termination) of a GP call. If the new bandwidth $y_{\tilde{s}}$ will be reserved for the new GP user $\tilde{s}$, the BE traffic rates x_r^o and price p_r^o ($r \in R_{BE}$) will move to the new optimal values $\tilde{x}_r^o$ and $\tilde{p}_r^o$; then, the capacity constraints in (2) become: $\boldsymbol{A}^{BE} \cdot \boldsymbol{x} \le \tilde{\boldsymbol{c}}$, where $\tilde{\boldsymbol{c}} = [\tilde{c}_j, j \in J]$ is the residual capacity matrix:

$$\tilde{c}_j = \begin{cases} c_j - \sum_{s \in R_{GP}(j)} y_s - y_{\tilde{s}}, & \text{if } j \in J(\tilde{s}) \\ c_j - \sum_{s \in R_{GP}(j)} y_s, & \text{if } j \notin J(\tilde{s}) \end{cases} \quad (5)$$

In this situation, since the $v_{\tilde{s}}$ can be interpreted as the maximum price accepted by the GP user $\tilde{s}$, the SP yields the maximum revenue contribution from the user by imposing the tariff $p_{\tilde{s}} = v_{\tilde{s}}$. Clearly, the assignment does not necessarily imply a more lucrative revenue equilibrium for the SP than the one achieved before the GP user $\tilde{s}$ has entered the network.

If a perfect knowledge of the users' utility functions is available, a pricing-based CAC policy could be applied to the GP calls, aimed at maximizing the SP's revenue:

$$\sum_{s \in R_{GP}} v_s \cdot y_s + v_{\tilde{s}} \cdot y_{\tilde{s}} + \sum_{r \in R_{BE}} \tilde{p}_r^o \cdot \tilde{x}_r^o \ge \sum_{s \in R_{GP}} v_s \cdot y_s + \sum_{r \in R_{BE}} p_r^o \cdot x_r^o \quad (6)$$

The new GP user $\tilde{s}$ is accepted in the network if (after a check on the bandwidth availability on the assigned routing path), it increases the SP's revenue derivative (e.g., $v_s \cdot y_s$ and $p_r^o \cdot x_r^o$ are expressed in €/minute). Such a strategy constitutes an upper bound for the revenue performance, since it is difficult to think at GP users declaring beforehand their utility functions to the SP.

Therefore, if the GP users' utility functions are unknown, the SP can fix the GP price $p_{\tilde{s}}$ as:

$$p_{\tilde{s}} = \sum_{r \in R_{BE}} \left[p_r^o \cdot x_r^o - \tilde{p}_r^o \cdot \tilde{x}_r^o \right] \quad (7)$$

in order to select only the GP users that increase the current revenue derivative. On the other hand, the GP user $\tilde{s}$ accepts to enter the network only if $v_{\tilde{s}} \ge p_{\tilde{s}}$.

In both cases, every time the CAC block acts, it is necessary to foresee the new revenue rate (denoted in the following with $\tilde{\Phi}_{BE} = \sum_{r \in R_{BE}} \tilde{p}_r^o \cdot \tilde{x}_r^o$), which is received by the BE traffic after the bandwidth reallocation for the new GP user $\tilde{s}$ has been provided.

This requires the on-line computation of the solution of a constrained non-linear mathematical programming problem. Actually, the key idea to perform the revenue forecast is to use the problem (2) to compute the new BE state vector $\tilde{\boldsymbol{x}}^o$ after the bandwidth reallocation; then, using (3), it is possible to calculate the new BE prices ($\tilde{p}_r^o = U_r^{'}(\tilde{x}_r^o)$, $\forall r \in R_{BE}$) and to evaluate the total revenue after the possible bandwidth reallocation.

III. THE DISTRIBUTED CONTROL ALGORITHM

We now investigate a decentralized control model to solve the aforementioned revenue forecast problem. As previously pointed out, the major concern comes from the need of computing the solution of (2) in order to forecast the new equilibrium $\tilde{\Phi}_{BE}$ after the possible acceptance of a new GP call. This reveals to be computationally expensive for large networks. We verified by simulation inspection that the related computational time is exponential in the number of users. To avoid the centralized computation, we take a closer look into the parameters involved in the pricing structure and in the network performance, thus giving rise to a proper approximating structure of the revenue forecast.

To avoid notational burden and without loss in generality, we suppose that both the GP and BE traffics are carried along K different routing paths. Each of them is managed by a single CAC *Decision Maker* (DM). Each DM knows the *state* of its path, defined as the maximum amount of bandwidth available for the BE users routed along its path, together with the BE users' willingness to pay. It periodically receives the information about the state of the other network paths from the other DMs, and approximates the effect (for the overall network) of the acceptance/rejection of the GP calls (routed along its path) as follows.

Let BE_k be the set of BE users routed along the *k*-th route in which CAC decisions are taken by the *k*-th DM. Let $\zeta_{BE}^k(t)$ be the available bandwidth for BE_k users at time *t*, namely, the bandwidth left unused by the GP traffic in the bottleneck link of route *k*. Let $\boldsymbol{m}_{BE}^k(t)$ be a set of parameters describing the overall willingness to pay of the BE_k users at time *t*. For instance, $\boldsymbol{m}_{BE}^k(t)$ could be some parameters describing the form of the utility functions. $\boldsymbol{m}_{BE}^k(t)$ could also describe the willingness to pay of BE_k users on an average basis, thus avoiding the need to know all BE users' utility functions in real time, by exploiting for example an off-line forecast, possibly together with an estimate of the current BE traffic demand. For the time being, it is important to highlight the information exchange process among the DMs. Let

$$\begin{aligned}\boldsymbol{I}_k(t) = col\{\ & \zeta_{\mathrm{BE}}^1(t-T),\ \boldsymbol{m}_{\mathrm{BE}}^1(t-T),\ \ldots \\ & \ldots\ \zeta_{\mathrm{BE}}^{\mathrm{k}}(t),\ \boldsymbol{m}_{\mathrm{BE}}^{\mathrm{k}}(t),\ \ldots \\ & \ldots\ \zeta_{\mathrm{BE}}^{\mathrm{K}}(t-T),\ \boldsymbol{m}_{\mathrm{BE}}^{\mathrm{K}}(t-T)\ \}\end{aligned} \tag{8}$$

be the information vector available for the *k*-th DM at time *t*, where T is the propagation delay of the information exchanged by the DMs. Let Ξ be the cardinality of such information vector.

When no knowledge of the BE users utility functions is available, the composition of the information vector (8) is only based on the BE *bandwidth availability* variables $\zeta_{BE}^k(t),\ k = 1,\ldots,\mathrm{K}$.

Each DM periodically broadcasts the state of its network path to the other DMs (i.e., the current values of the vectors ζ_{BE}^k, $\boldsymbol{m}_{BE}^k$). If a new GP call arrives at time *t* for the *k*-th route, the *k*-th DM infers its potential effect on the future network performance (in terms of $\tilde{\Phi}_{BE}$) on the basis of its information vector $\boldsymbol{I}_k(t)$ and, then, performs the CAC decisions through (6) or (7). Clearly, to solve (6) or (7), each DM must receive the information related to the current GP income deriving from the other network paths, too.

Let now $\gamma_k(\boldsymbol{I}_k(t)) : \Re^{\Xi} \to \Re^+$ be the decision function of the *k*-th DM and, without loss in generality, let $\gamma_k(\boldsymbol{I}_k(t))$ act on the revenue performance, namely, the revenue forecast is computed by the DM through $\tilde{\Phi}_{BE}(t) = \gamma_k(\tilde{\boldsymbol{I}}_k(t))$. We denote with $\tilde{\boldsymbol{I}}_k(t)$ the information vector whose $\zeta_{BE}^k(\cdot)$ variables are updated in coherence with the new capacity constraints $\tilde{\boldsymbol{c}}$ (5), as if the incoming GP call $\tilde{s}$ were accepted in the network. We call the functions $\gamma_k(\boldsymbol{I}_k(t)),\ k = 1,\ldots,\mathrm{K}$ *Revenue Forecast Functions* (RFFs).

The **Decentralized Control Problem (DCP)** addressed in this work can now be stated. It consists in finding the *Optimal RFFs* $\gamma_k^o(\cdot)\quad k = 1,\ldots,\mathrm{K}$ in order to compute the revenue forecast as if a centralized computation of $\tilde{\Phi}_{BE}$ (through (2) and (3)) were feasible. The DCP formulated here is a *functional optimization* problem, since the optimal form of the functions $\gamma_k(\cdot)$ is under investigation to mimic a centralized management of the network. The solution of such functional optimization problem through analytical tools is a very hard task. Only numerical approximations can be provided.

In order to approximate the solution of such DCP, we make use of *one hidden layer feedforward neural networks*, denoted in the following with $\hat{\gamma}_k(\boldsymbol{I}_k(t), \boldsymbol{\omega}_k),\ k = 1,\ldots,\mathrm{K}$, where $\boldsymbol{\omega}_k$ is the vector of the *k*-th neural network's weights. We call the approximating functions $\hat{\gamma}_k(\cdot, \boldsymbol{\omega}_k) : \Re^{\Xi} \to \Re^+$ *Neural Revenue Forecast Functions* (Neural RFFs). Each $\hat{\gamma}_k(\cdot, \boldsymbol{\omega}_k)$ can be trained in order to approximate the *Optimal RFFs* above, by formulating the following non-linear mathematical programming problem.

The **Neural Revenue Forecast Functions Training (NRFFT)** is aimed at finding the optimal neural network's weights assignment $\boldsymbol{\omega}_k^o$, so that:

$$\sum_{h=1}^{H} \left[\Phi_{BE}^h - \hat{\gamma}_k^h(\boldsymbol{I}_k^h, \boldsymbol{\omega}_k^o) \right]^2 \le \xi,\ k = 1,\ldots,\mathrm{K} \tag{9}$$

The problem NRFFT consists of tuning the *k*-th neural network's outputs $\hat{\gamma}_k^h(\boldsymbol{I}_k^h, \boldsymbol{\omega}_k),\ h = 1,\ldots,H$ for $\forall k$, in order to approximate the collected values of the BE revenue $\Phi_{BE}^h,\ h = 1,\ldots,H$ as function of the samples of the state of the network, collected in the information vector instances $\boldsymbol{I}_k^h$, $h = 1,...,H$.

The index h denotes the h-th component of a training set (the couple $\boldsymbol{I}_k^h$ - Φ_{BE}^h) obtained in the course of a simulation phase in which both the *Neural RFFs* $\hat{\gamma}_k(\cdot, \boldsymbol{\omega}_k)$ and (2)-(3) are applied to compute the terms Φ_{BE}^h and $\hat{\gamma}_k^h(\cdot)$ in (9). Typical values for the bound ξ are in the range [0.5;0.01]. The problem NRFFT states a regular neural network's training problem.

In this way, the functional optimization DCP has been reduced to a non-linear mathematical programming problem, easily solvable by conventional mathematical programming algorithms. If the training set $h = 1,...,H$ is sufficiently large and the *Neural RFFs* are equipped with an adequate number of neural units, due to the well-known generalization properties of neural networks, it is expected that the optimized *Neural RFFs* $\hat{\gamma}_k(\cdot, \boldsymbol{\omega}_k^o)$ is capable to infer the correct BE revenue forecast $\tilde{\Phi}_{BE}$ with respect to any possible configuration of the information vector (not only in relation to the instances collected in the training set of (9)).

The training procedure related to the NRFFT problem can be performed off line. In this respect, the required computational burden does not influence the on-line performance of the DMs. In real time, the trained RFFs can be applied in a decentralized fashion "almost instantly", thus avoiding any on-line computational burden.

IV. Simulation Results

In this Section we illustrate an evaluation of our pricing-based CAC mechanisms. To this aim, we have developed a simulation tool in C++ that describes the behaviour of the network at call (GP traffic) and flow (BE traffic) level. The *COST 239* experimental network (depicted in Fig. 1) is utilized for the tests; it is composed of 20 links and 11 nodes. We consider a subset of 6 active routes, where each active route can generate both BE and GP traffic connections:

Route 1: { 0, 1, 4, 8 } ; Route 2: { 4, 8, 6 } ; Route 3: { 9, 12, 19 } ;

Route 4: { 4, 8, 11, 16 } ; Route 5: { 9, 12, 14 } ; Route 6: { 5, 6, 9 } ;

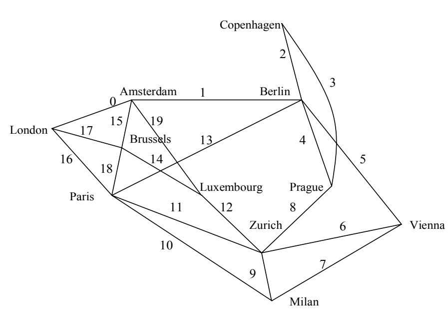

Fig. 1. Topology of the test network.

For the results presented, the width of the confidence interval over the revenue performance is less than 1% of the estimated value for 95% of the cases. A constrained non-linear programming solver is integrated in the simulator to approximate the behaviour of the BE portion of the network and to deploy (6) and (7) in a centralized way.

We use the following utility functions for both the BE and the GP traffic:

$$U_r(x_r) = \alpha_{BE}\sqrt{x_r}\,;\; x_r \in [0.1;1.0]\,;\forall\; r \in R_{BE} \qquad (10)$$

$$U_s(y_s) = \alpha_{GP}\sqrt{y_s}\,;\; y_s \in [0.1;1.0]\,;\forall\; s \in R_{GP} \qquad (11)$$

where the parameters α_{BE} and α_{GP} are random variables generated from a uniform distribution to describe variable willingness to pay conditions. The average willingness to pay of the GP users is fixed equal to the BE users' one by setting $\bar{\alpha}_{GP} = \bar{\alpha}_{BE} = 1.0$.

The simulation data are summarized as follows: **1)** call frequency: $\lambda_k^{BE} = \lambda_k^{GP} = \lambda = 1.0$ call per minute (exponentially distributed), $\forall$ route $k \in$ {1, …, 6}; **2)** average call duration: $\frac{1}{\mu_k^{BE}} = \frac{1}{\mu_k^{GP}} = \frac{1}{\mu} = 10.0$ minutes (log-normally distributed), $\forall$ $k \in$ {1, …, 6}; **3)** $\bar{\alpha}_{BE}^k = \bar{\alpha}_{BE} = \bar{\alpha}_{GP}^k = \bar{\alpha}_{GP} = 1.0$, $\forall$ route $k \in$ {1, …, 6}; **4)** link capacity : $c_j = c = 30.0$ Mbps, $\forall$ link $j \in$ {0,…,19}; **5)** time horizon of the simulation: 10.0 hours.

The revenue and the blocking probability performance achieved by all the aforementioned CAC policies are depicted in Figs. 2 and 3. In order to limit the notational burden, we indicate with "*.GPiBEj*" *i*=1,0, *j*=1,0 the adoption of a CAC policy with *i*=1, if an on-line knowledge on GP users' utility functions is available (*i*=0 otherwise), with *j*=1, if an on-line knowledge on BE users' utility functions is available, and with *j*=0 otherwise. For instance, the centralized application of (6) and (7) is denoted by "*Centralized.GP1BE1*" and "*Centralized.GP0BE1*", respectively.

It is easily observable from Fig. 2 that the proposed centralized control, with a feedback on the users' utility functions (i.e., *Centralized.GP1BE1*), succeeds in yielding the best revenue performance. The performance reached by the other centralized policy (*Centralized.GP0BE1*) is much lower. The revenue percentage decrease is around 36%.

The blocking probability values with respect to only the bandwidth availability check are below 0.8%. It means that most of the contribution to the overall blocking probability comes from the application of the proposed pricing-based comparisons (6) and (7). The blocking probability obtained by the *Centralized.GP0BE1* technique (22.5%) is quite high with respect to the ideal value guaranteed by the *Centralized.GP1BE1* strategy (5.6%) (see Fig. 3).

As far as the decentralized control is concerned, we must note that a single DM is associated to the each network route of Fig. 1. Each DM implements the CAC (7), since no knowledge of the GP users' utility functions is exploited when our decentralized control is put in practice. The *Neural RFFs* are implemented by a feedforward neural network with 100 hyperbolic tangent neural units in the hidden layer and with a linear output layer. The training phase was performed with up to 10,000 samples of the information vectors (8) (and corresponding BE revenue values) taken from the repetition of the proposed simulation scenario until the training performance requirement $\xi = 0.05$ in (9) had been reached. We used the *Resilient Backpropagation* algorithm within the *Matlab 6.5* environment after producing the training samples through the C++ simulator.

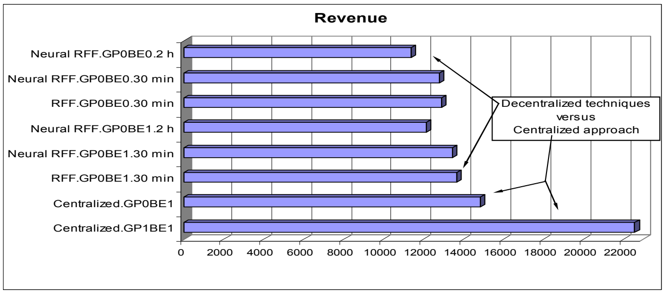

Fig. 2. Revenue Performance.

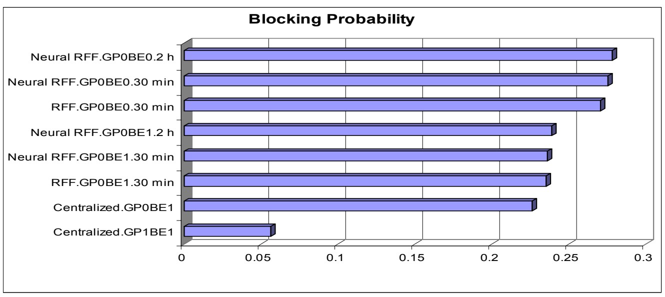

Fig. 3. Blocking Probability Performance.

The overall willingness to pay of the BE users routed along the *k*-th routing path, m_{BE}^{k}, is described in the DMs' information vectors $\boldsymbol{I}_k$, $k = 1,...,6$ as the sum of the parameters defining the square root utility functions in (10), i.e., $m_{BE}^{k}(t) = \sum_{r \in R_{BE}^{k}(t)} \alpha_{BE}^{k}$, where $R_{BE}^{k}(t)$ denotes the set of all BE_k users (characterized by square root utility functions) active at time *t* on the *k*-th routing path. This composition of the information vector defines the decentralized strategies denoted with "*.GP0BE1*".

The results shown in Figs. 2 and 3 are obtained by averaging the *Neural RFFs* along 10 independent replications of the proposed simulation scenario, different from the ones adopted during the training phase.

We denote with "*Neural RFF.T*" the application of the *Neural RFFs* where a specific time interval *T* (in minutes or hours) is necessary for each DM to receive a stable update about the state of the other network paths.

The "*RFF*" notation denotes the application of the decentralized control policy (7), i.e. on the basis of the DMs' information vectors, but without applying the neural network resulting from the solution of problem NRFFT. The idea is to solve (7) in a decentralized way, but only with the information vector available by the DM performing the CAC decision. Using such information vector, we compute the revenue forecast as done for the centralized techniques (instead of applying the trained neural network). In this way, it is possible to highlight the network performance as function of the decentralized information structure, without taking into account the estimation error introduced by the neural approximation. Looking at Fig. 2, it is easily observable that the performance difference between the *Neural RFF.T* and the *RFF.T* policies highlights that such estimation error is quite low. This suggests that the application of the neural network produces a good approximation of the BE revenue forecast $\tilde{\Phi}_{BE}$.

Looking at Fig. 2, it is clear that the proposed decentralized control well approximate the performance achieved by the *Centralized.GP0BE1* technique, even when no knowledge of BE users' utility functions is available for the DMs. The overall percentage decrease is around 8%.

One final remark is necessary to comment the decentralized control's performance with respect to the propagation delay *T*. A low performance decrease is obtained, despite the increase in *T* (from 30 minutes to 2 hours, see Fig. 2). This effect corroborates the robustness of the decentralized control for the investigated network environment.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a novel pricing-based CAC problem has been investigated in relation to the decentralized management of the network. Such decentralized control has been obtained by reducing the CAC formulation to an approximating scheme suitable for neural network training. The performance evaluation confirmed the good performance of the proposed approach.

Future work will regard the application of the proposed control methodology to *path restoration* and *routing balancing* optimization problems (following also the "relaxation" method adopted in [5]), typically requiring mixed-integer programming formulations and a centralized management of the network.

REFERENCES

[1] C. M. Lagoa, H. Che, B. A. Movsichoff, "Adaptive Control Algorithms for Decentralized Optimal Traffic Engineering in the Internet," *IEEE/ACM Trans. on Networking*, vol. 12, no. 3, June 2004, pp. 415-428.

[2] C. Courcoubetis, R. Weber, *Pricing Communication Networks - Economics, Technology and Modelling*, John Wiley and Sons, Etd., Chichester, UK, 2003.

[3] F.P. Kelly, A.K. Maulloo, D.K.H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *J. of Operat. Res. Soc.*, vol. 49, no. 3, May 1998, pp. 237-252.

[4] S.H. Low, D.E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, Dec. 1999, pp. 861-874.

[5] C. Y. Wei, M. Naraghi-Pour "Path restoration with QoS and Label Contraints in MPLS Networks," *Proc. IEEE Internat. Conf. on Commun. (ICC 2004)*, Paris, July 2004.