A Measurement-Based Adaptive Control Mechanism for Pricing in Telecommunication Networks

Franco Davoli, Mario Marchese, and Maurizio Mongelli

Abstract: **The problem of pricing for a telecommunication network is investigated with respect to the users' sensitivity to the pricing structure. A functional optimization problem is formulated, in order to compute price reallocations as functions of data collected in real time during the network evolution. No a-priori knowledge about the users' utility functions and the traffic demands is required, since adaptive reactions to the network conditions are sought in real time. To this aim, a neural approximation technique is studied to exploit an optimal pricing control law, able to counteract traffic changes with a small on-line computational effort. Owing to the generality of the mathematical framework under investigation, our control methodology can be generalized for other decision variables and cost functionals.**

Index Terms: **Functional optimization, network pricing, neural control, user sensitivity.**

I. INTRODUCTION

Network pricing is an issue widely treated in the literature. In the last decade, quite a few models have been proposed to address the network management through the pricing structure [1]. In this paper, we pursue the calculation of an optimal pricing control law as a function of the users' responsiveness to the pricing structure, by exploiting data collected in real time during the system's evolution. We investigate a pricing model suitable for guaranteed performance (GP) users (i.e., requiring specific quality of service (QoS) constraints), sharing resources with best effort (BE) ones. In the literature, GP and BE pricing mechanisms are usually analyzed separately. Both of them have their specific advantages and drawbacks. BE pricing is related to elastic (TCP) traffic; it is based on network congestion feedback, and it exploits a decentralized algorithm (integrated with flow control) to assure users' welfare maximization, with a small computational effort [2], [3]. On the other hand, it disregards any revenue consideration and does not offer support to QoS-based services. It is based on the concept of utility function to abstract users' responsiveness to prices. Since utilities are difficult to estimate, there are no effective instruments for the service providers (SPs) to control the network's evolution. As far as GP pricing is concerned, it considers inelastic traffic with QoS guarantees [4]–[8]; it offers an explicit support to influence network evolution, because it is possible for SPs to decide prices and bandwidth allocations. However, it requires a centralized management of the network with a significant computational effort. Users' responsiveness to prices is supposed to be known a priori, too. This work investigates a pricing control algorithm that tries to cope with all the drawbacks mentioned above, and explicitly considers multiplexing BE and GP traffic flows within a single multiservice network. The remainder of the paper is organized as follows. In the next section, we shall start with an insight into the BE and GP pricing models in order to motivate our approach. In Section III, we define both the network model and the revenue maximization problem under different stochastic environments and we highlight the need of investigating an adaptive pricing control mechanism. In Section IV, we formulate our functional optimization approach and, in Section V, we investigate a neural approximating technique to solve our problem. In Section VI, we validate our methodology through simulation analysis. In Section VII, we conclude, by summarizing the obtained results and emphasizing the directions for future research.

Manuscript received May 21, 2008; approved for publication by Song Chong, Division III Editor, November 16, 2009.

The authors are with the Department of Communications, Computer, and Systems Science, University of Genova, Genova, Italy, email: {franco, dama, mopa @dist.unige.it}.

II. STATE OF THE ART AND OPEN ISSUES

A. BE and Utility-Based Pricing

Several works exploit the representation of users' satisfaction through utility functions. The concept of *utility function* was introduced in the telecommunication literature to depict the QoS as appreciated by the users, mainly by associating a higher degree of QoS, and hence of user satisfaction of a service, to the amount of bandwidth available for that service. Thus, it is possible to define the utility of a user r as a function of the user's bandwidth assignment x_r, namely $U_r(x_r)$. Such function describes how sensitive user r is to changes in x_r. In the context of pricing, it is useful to think of it in terms of the amount of money user r is willing to pay for a cer tain x_r. Let a telecommunication network be composed by a set J of unidirectional links and a set R of users (source-destination (SD) pairs). Link j has capacity c_j, $J(r)$ is the subset of J containing the links traversed by user r, $R(j)$ is the subset of users traversing link j. Let $\mathbf{A} = \{A_{jr}, j \in J, r \in R\}$ be the matrix assigning resources to users (A_{jr}=1 if link j is traversed by user r's traffic, A_{jr}=0, otherwise). In such a context, firstly formulated in [2], each user accessing the network maximizes his/her net utility with respect to the assigned price p^r, i.e., the bandwidth demand x_r is ruled by (1) below (m_r and M_r denote the lower and upper limits of the bandwidth domain, respectively):

$$x_r^o = \arg\max_{x_r \in [m_r, M_r]} [U_r(x_r) - x_r p^r] \quad (1)$$

with

$$p^r = \sum_{j \in J(r)} \{p_j\}. \quad (2)$$

If we interpret p_j as the price per unit bandwidth at link j, then p^r is the total price per unit bandwidth for all links in the path

1229-2370/10/$10.00

of user r. Hence, $x_r p^r$ represents the *shadow price* for user r, namely, the bandwidth cost when transmitting at rate x_r, under the current network conditions (in terms of the congestion of the links and of the other users' utility functions). We now briefly recall the results of [2] and [3], which show that the very same choice of x_r maximizes the so-called network *social welfare* when all users react to prices as outlined in (1), i.e.,

$$\boldsymbol{x}^o = \underset{\substack{x_r \in [m_r, M_r], r\in R,\\ \mathbf{A}\cdot\boldsymbol{x}\leq\mathbf{c}, \boldsymbol{x}\geq 0}}{\arg\max} \sum_r U_r(x_r) \tag{3}$$

where $\mathbf{A}$, $\boldsymbol{x}$, and $\mathbf{c}$ are the aggregate vectors of the assigned resources, users' rates, and link capacities, respectively. Moreover, a decentralized congestion-dependent implementation of such pricing scheme is available. In brief, the key idea is to exploit the Lagrangian decomposition of (3), thus giving rise to a flow control mechanism of the form [3]:

$$x_r(p^r(t)) = [U_r^{'-1}(p^r(t))]_{m_r}^{M_r} \tag{4}$$

$$p_j(t+1) = p_j(t) - \eta \left\{ c_j - \sum_{r\in R(j)} \{x_r(p^r(t))\} \right\} \tag{5}$$

where $x_r(p^r(t))$ is the solution of (1); $[z]_a^b$=min$\{\max\{z,a\},b\}$, $U_r^{'-1}$ denotes the inverse of the utility function's derivative, and η is the gradient step-size. At each iteration, user r individually solves (1) through (4) and sets the rate on the respective SD path $J(r)$ to $x_r(p^r(t))$. Then, each link $j \in J(r)$ updates its price p_j according to (5), it communicates the new prices to users $r \in R(j)$, whose transmission rate must be changed according to (4), and the cycle repeats. Such pricing mechanism is integrated within the flow control and achieves an ideal situation in which all users act individually, by pursuing their own benefits, but, at the same time, by guaranteeing the maximization of the network welfare (3). This mechanism is appropriate for contracts with flexible guarantees, related to "elastic" applications. The most important drawbacks of such pricing model are the following:

i) The revenue metric is not considered;
ii) prices are defined with respect to the current level of congestion (see (5)), disregarding any QoS constraint;
iii) the SP's perception of utility functions is very limited.

Even if some works investigate the user responsiveness to the perceived QoS and the tariff structure (see e.g., [9]–[11]), the notion of utility is actually difficult to measure or estimate. Utility-based pricing is studied as if utility functions could be a priori known for the SP, which optimally tunes the call admission control (CAC) as if each user device could declare its utility function in advance [12], [13]. In practice, when the flow control mechanism (4)–(5) takes place, the SP does not know the users' utility functions and does not have any direct control on the prices of the BE users. Then, the only way for the SP to control the BE prices is to influence the state of the network congestion. By applying specific routing strategies, it can introduce fictitious points of congestion in order to increase the values of the prices in (5), but it cannot control explicitly the equilibrium point of the BE portion of the network. In practice, another approach for the optimization of network pricing is possible and it is related to the GP traffic type.

B. GP Pricing and Traffic Demands' Sensitivity

Real time traffic complicates the situation even more, since it requires QoS guarantees, and gives rise to a pricing structure involving the corresponding *effective bandwidth* of the services [1] and non-concave utility functions (which do not guarantee the existence of a unique optimum of (3)). In this perspective, it is difficult to think of prices tracking the statistical fluctuation of congestion, as in (4) and (5), whereas it is more realistic to deploy fixed prices (known in advance by the users) or time-dependent ones, according to slowly varying parameters that capture the prevailing operating conditions of the system [4], [14]. The user's responsiveness to the tariff structure is thus related to the interarrival of the connection requests, disregarding any utility-based consideration. For each class of service, in which QoS requirements are guaranteed in an equivalent bandwidth fashion, [4], [8], [14], [15] define frequency functions of the service requests $\lambda(\cdot)$ with respect to the assigned prices, namely: $\lambda(\cdot) = \lambda(p)$. In [5], a somehow similar approach is proposed, where $\lambda(p)$ is the packet arrival rate of BE traffic as a function of the price p. In the presence of multiple (K) service classes, let $\boldsymbol{\lambda}(\boldsymbol{p})$ be the aggregate vector of the traffic laws $\lambda_\kappa(p_k)$; $\kappa = 1,\cdots,K$. Also in this case, the maximization of the network performance still remains an open issue, since different choices on the prices give rise to different evolutions of the system. It is possible to exploit proper mathematical instruments for the planning of the telecommunication network, but some severe drawbacks still remain unsolved.

i) The first one is related to the computational burden involved in such mathematical tools (*dynamic programming* in [4]–[7], [14], [15], *mixed integer mathematical programming* in [8]), which limits their application if real time reactions are needed. The adoption of some self-adjustment mechanism to support on-line prices' reallocation in the presence of variable system conditions [16] is needed.
ii) The second (and most important) one regards the assumption made on the perfect knowledge of the traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$. Some knowledge on the user's responsiveness to the pricing structure is supposed to be always in effect in [4], [5], [8], [14], [15]. If a perfect knowledge of users' utility functions is difficult to assure in a real context, the same holds true for the estimate of the functions $\boldsymbol{\lambda}(\boldsymbol{p})$ [4], [14].
iii) Moreover, as underlined in [4], [8], it is worth noting that the optimal price allocation $\boldsymbol{p}^o$ can be only obtained through a centralized management of the network.
iv) Finally, the effect of time-varying bandwidth allocations (typical of BE services) is not taken into account if only the traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$ of the GP users are considered.

C. The Present Approach

In virtue of the points outlined above for both BE and GP pricing, optimizing the pricing structure, jointly with on-line estimations of actual traffic demands, still deserves attention [17]. The idea of the present work is thus to formulate a novel pricing control algorithm, such that:

i) It infers the optimal prices as functions of measures obtained in real time, without any on-line knowledge of the functions $\boldsymbol{\lambda}(\boldsymbol{p})$ and $\boldsymbol{U}(\cdot)$ ($\boldsymbol{U}(\cdot)$ being the aggregate vector of the utility functions);

ii) it reacts on line to non-stationary $\boldsymbol{\lambda}(t, \boldsymbol{p})$ and $\boldsymbol{U}(t, \cdot)$;
iii) it manages both GP and BE traffic, multiplexed together and sharing the available resources;
iv) it avoids any on-line computational burden;
v) it is suitable for a decentralized control of the network. Together with the GP traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$ (as in [4], [8], [14], [15]), we exploit utility functions $\boldsymbol{U}(\cdot)$.

We suppose that a BE user r reacts to variations in the rate allocation by updating the congestion control scheme as outlined in (4), and that a GP user s refuses to subscribe a GP traffic contract, if the imposed price p_s does not satisfy the *willingness to pay* $\nu_s = U_s^{'}(y_s)$ for the required rate y_s (obtained through (1) by replacing x_r with y_s and p^r with ν_s). If $p_s \geq \nu_s$, the GP user s, with required rate y_s, refuses to enter the network. We investigate a novel optimization model for the pricing structure. The functional optimization approach underlying the pricing allocation is exploited, with respect to the state of the system. We take the SP's revenue as the final goal to pursue.

III. THE NETWORK MODEL AND THE REVENUE OPTIMIZATION PROBLEM

We start by considering the GP traffic only. With a notation that slightly differs from [8], let us consider H traffic routes within a telecommunication network. A route $h \in \{1, \cdots, H\}$ is defined as a network path assigned to a group of GP users, according to the required source-destination nodes and with respect to the chosen routing plan. For each route, K different QoS treatments are available. A *service class* is identified in terms of assigned route, QoS treatment and assigned price $p_{h\kappa}$. In the multiprotocol label switching (MPLS) terminology, a service class is equivalent to the concept of forwarding equivalent class (FEC), established on a specific virtual path (VP). The corresponding equivalent bandwidth requests are denoted with $y_{h\kappa}$ and the corresponding traffic laws with $\lambda_{h\kappa}(p_{h\kappa})$. For instance, p may be in terms of [\$ per Mbps per minute] and y in [Mbps]. Following [8], a *service separation* among the service classes [18] is implemented in each network node. This means that a buffer is provided for each class and a scheduler is supposed to guarantee a proper bandwidth allocation among the classes. By exploiting the traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$, different network behaviors are possible in terms of shared resources and corresponding network performance (blocking probability, revenue, welfare, and so on). Such performance metrics are manageable by the SP, by implementing a proper tariff structure $\boldsymbol{p}(t) = col\,\{p_{h\kappa}(t); h = 1, \cdots, H; \kappa = 1, \cdots, K\}, t \in [0, +\infty]$. Disregarding, for the time being, any utility-based consideration, the first pricing problem is formulated as follows. **Pricing allocation problem** (PAP) **I**: Find the optimal tariff assignment $\boldsymbol{p}^o(t)$, $t \in [0, +\infty]$, in such a way that the long-term average SP's revenue defined in (6) below is maximized:

$$\boldsymbol{p}^o(t) = \arg\max_{\boldsymbol{p}(t)} E_{\boldsymbol{\omega}} L[\boldsymbol{p}, \boldsymbol{\omega}];$$

$$L[\boldsymbol{p}, \boldsymbol{\omega}] = \lim_{T \to \infty} \frac{1}{T} \int_0^T \sum_{h=1}^{H} \sum_{\kappa=1}^{K} n_{h\kappa}(\tau) p_{h\kappa}(\tau) y_{h\kappa}(\tau)\, d\tau \quad (6)$$

where $n_{h\kappa}(t)$ is the number of active users of service class $h\kappa$ at time t, and $\boldsymbol{\omega}$ represent a sample path of all stochastic variables involved in the generation of $\{n_{h\kappa}(t); h = 1, \cdots, H; \kappa = 1, \cdots, K\}$, $t \in [0, +\infty]$. The tariff assignment influences the stochastic variables $n_{h\kappa}(t)$ through the (known) traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$, i.e., by influencing the arrival rate of users' connection requests (according to their specific statistics, generally Poissonian). This is one of the stochastic elements determining the sample path $\boldsymbol{\omega}$; the others are the statistics of connection durations and the distribution of connections among traffic classes (with corresponding bandwidth requirements). Thus, each $n_{h\kappa}(t)$ depends on the chosen tariff structure $\boldsymbol{p}(\tau)$ (and corresponding arrival rates $\boldsymbol{\lambda}(\tau)$, $\tau \in [0, t]$) and on how the service classes have shared the available bandwidth under the assigned routing paths in the time interval $[0, t]$. For each new GP request, a CAC agent is supposed to guarantee the network constraints due to the limited bandwidth of the links, thus rejecting the incoming call in case of insufficient resources. We include in $\boldsymbol{\omega}$ all the stochastic variables involved in the system, i.e., the aggregate vectors of the arrival process of connection requests $(\boldsymbol{a})$, of the call durations $(\boldsymbol{d})$ and of the bandwidth requirements $(\boldsymbol{y})$, which can follow complex interactions among the traffic sources through statistical multiplexing, i.e., $\boldsymbol{\omega} = [\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{y}]$. A similar formulation of PAP I is outlined in [6]. In general, functions $\boldsymbol{p}(t)$, $t \in [0, +\infty]$ may represent either "open-loop" or "closed-loop" control laws (in the latter case, $\boldsymbol{p}(t) = \boldsymbol{\psi}[\boldsymbol{n}(t)]$, $\boldsymbol{n}(t)$ being the vector of the number of active users of all service classes). Finding them may be a formidable problem. A description of the users' sensitivity with respect to a utility-based approach is also possible. This leads to traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$ slightly different from the ones above. If a utility function $U(y)$ is introduced to depict the user's satisfaction with respect to the rate assignment y, the calls coming from each service class follow a randomly modulated process with rate $\lambda_{h\kappa}^o \boldsymbol{P}[U_{h\kappa}^{'}(y) \geq p_{h\kappa}]$, where $\lambda_{h\kappa}^o$ is the arrival rate when the price is zero ([14]). The SP's expected long-term average revenue reported in (7) below defines the **pricing allocation problem** (PAP) **II**:

$$L[\boldsymbol{p}, \boldsymbol{\omega}] = \lim_{T \to \infty} \frac{1}{T} \int_0^T \sum_{h=1}^{H} \sum_{\kappa=1}^{K} p_{h\kappa}(\tau) y_{h\kappa}(\tau) \left|\Phi_{h\kappa}^{GP}(t)\right| dt \quad (7)$$

where $\left|\Phi_{h\kappa}^{GP}(t)\right|$ is the subset of active users of service class $h\kappa$ at time t, whose willingness to pay prevails over the assigned tariff $p_{h\kappa}(t)$, namely $U_\varsigma^{'}(y_\varsigma(t)) \geq p_{h\kappa}(t)$, $\varsigma \in \Phi_{h\kappa}^{GP}(t)$. $|\Phi|$ denotes the cardinality of the set Φ. The possible sample paths of the system depend now on the utility functions, too.

Finally, let $l_{GP}(t) = \sum_{h=1}^{H} \sum_{\kappa=1}^{K} n_{h\kappa}(t) p_{h\kappa}(t) y_{h\kappa}(t)$ be the overall revenue per time unit available at time t under a possible realization of the stochastic processes with respect to the GP users' behavior under the traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$ (denoted now with ${}^{GP}\boldsymbol{\lambda}(\boldsymbol{p})$). We consider now also the presence of BE traffic, by taking PAP I stated in (6) as a reference. BE users obtain the bandwidth left unused by the GP ones. We suppose that the BE traffic follows a bandwidth allocation $\boldsymbol{x}(t)$, according to the flow control mechanisms (4) and (5), thus tracking the optimal solution of (3), as outlined in Section II. The rationale of this choice is due to the fairness in the bandwidth and price allocations attainable among the BE users [2]. Moreover, by exploiting proper utility functions, it is possible to mimic the TCP behavior (see e.g., [19] and references therein). The chosen control variables are again the prices $\boldsymbol{p}$ imposed to the GP users, disregarding any direct action over the BE traffic (though other pricing schemes are possible for the BE traffic and might be taken into account). The overall SP's revenue is thus obtained as:

$$L[\boldsymbol{p}, \boldsymbol{\omega}] = \lim_{T \to \infty} \frac{1}{T} \int_0^T [l_{GP}(t) + l_{BE}(t)]\, dt \qquad (8)$$

and defines the **pricing allocation problem** (PAP) **III**. We denote here the revenue per time unit generated by the BE portion of the network at time t, as $l_{BE}(t) = \sum_{r=1}^{|\Phi^{BE}(t)|} p^r(t)\, x_r(t)$ ($|\Phi^{BE}(t)|$ is the set of active BE users in the system at time t). The stochastic environment is now $\boldsymbol{\omega} = [{}^{GP}\boldsymbol{a}, {}^{GP}\boldsymbol{d}, \boldsymbol{y}, {}^{BE}\boldsymbol{a}, {}^{BE}\boldsymbol{d}, \boldsymbol{x}]$. When GP and BE users are multiplexed together, the maximization of the network performance is not so trivial as it might be expected and an optimal trade-off in the bandwidth sharing exists [12], [20]. The key idea is to find the best bandwidth sharing among the users, by exploiting the GP traffic demands ${}^{GP}\boldsymbol{\lambda}(\boldsymbol{p})$. In any case, the generality of the proposed technique would allow us to investigate also the application of other control variables.

A. Mathematical Aspects

An approach in terms of Markov decision processes (MDP) is used in general to formulate pricing allocation problems in telecommunication networks [4], [8], [14], [15]. Our formulations of PAPs I, II, and III do not follow this line in virtue of the following reasons. MDP modeling leads to a complex notation when dealing with pricing problems (see e.g., the discussion of Section 2.5 of [6]). As such, it limits the modeling power of specific applications of interest (e.g., PAPs II and III). The MDP model is often derived and validated via simulations for the single-link case only; the formal generalization to the multi-link case can be straightforward, but the application of the resulting algorithms may be not so immediate. Closed-form expressions of the exact MDP solutions cannot be obtained for general traffic models (i.e., outside Markovian hypotheses) [6]. The calculation itself of the solution is also computationally expensive, due to the "curse of dimensionality" of dynamic programming underlying the MDP methodology. Some model reduction techniques are used to avoid such a computational burden. At the best of the authors' knowledge, the real time application of MDP solutions is effectively adopted only in the presence of some "neural programming" approximation of the original MDP equations [21].

B. Technology Aspects

Optimization problems with revenue structures such as those in PAP I and PAP II ((6) and (7)) could be related to a market-based environment, in which the SP manages connection requests from a pool of users and only a subset of them accept the assigned price. A proper negotiation protocol between client and SP can be used to allow each client to choose on-line the profitability of the service and the SP to decide whether to provide bandwidth or whether to wait for more lucrative requests (see e.g., [22] and references therein). For instance, a web hosting application, developed by the company "*invisible hand networks*" [23], [24], allows flexible reservation and dynamic pricing through web hosting providers. Such a mechanism could be suitable for *peer-to-peer* markets (such as [25]) or for negotiating digital (e.g., television) services on demand. Other important applications to real systems are the ones considered in [17]: IBM [26] or Cisco [27] "service center" environments where signaling at application level supports service contracts in real time with end users. In these kinds of environments, the SP is expected to tune the prices on line, according to the market conditions. In this work, we face this situation, disregarding the aspects related to the competition among different SPs [22]. Beyond the proprietary solutions mentioned above, the recent standardization studies regarding the IP multimedia subsystem (IMS) architecture [28], made on the interoperability of Internet technology with cellular networks with flexible pricing structures [28], motivate even more research efforts in this direction.

C. Parameter Adaptive and Dynamic Pricing Strategies

Works [4], [14], [15] have shown that, in large telecommunication networks, if the statistics of the sources are quite "regular" (i.e., stationarity of the traffic demands $\boldsymbol{\lambda}(\boldsymbol{p})$), the performance guaranteed by an optimal *dynamic* pricing strategy (in which the prices are assigned as functions of the number of users currently active in the network) can be always reached by an optimal *static* pricing strategy, in which fixed prices are always in effect, independent of the state of the system. Such optimal static prices must be calculated by means of a proper off-line planning of the network, simply by evaluating the steady state of the system (see e.g., [4], [8], [14], [15]). This static policy, based on a perfect knowledge of the users' sensitivity, is applied typically over some time horizons of a day and, for this reason, such planning procedure is often called *time-of-day pricing* [14], [29]. As pointed out in [14], the adoption of an adaptive tariff assignment $\boldsymbol{p}^o$, as function of measures performed over the real system and without supposing any exact form of the traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$), is quite attractive, since the evaluation of such traffic laws is an open area of research (see e.g., [9], [10], [11] and references therein). Therefore, our aim is to find a way to dynamically adjust a time-varying, static pricing policy $\boldsymbol{p}(t)$ on the basis of the current users' sensitivity, without assuming any a-priori knowledge of both the traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$ and the utility functions $\boldsymbol{U}(\cdot)$. These functions are assumed to be time dependent, i.e.: $\boldsymbol{\lambda}(t, \boldsymbol{p})$, $\boldsymbol{U}(t, \cdot)$, thus leading to a non-stationary stochastic environment. In this way, the need of employing an adaptive pricing scheme arises as a consequence. On the other hand, if also the BE traffic is multiplexed with the GP one, a pricing strategy, with a feedback on the current state of the network, is necessary to optimize the system performance [12]. The proposed approach guarantees instruments for the aforementioned *time-of-day pricing* or even for faster reactions in front of a non-stationary environment.

IV. A FUNCTIONAL OPTIMIZATION APPROACH

To face the above-described problems, we develop a functional optimization approach, by defining pricing reallocations as functions of an *information vector*, which summarizes the "recent history" of the network. A decision maker (DM) is assigned to each service class $h\kappa$ ($DM_{h\kappa}$) to monitor its temporal behavior in real time and to assign the price $p_{h\kappa}$ accordingly.

Following the idea of our previous work [30], where a structure of DMs allows the decentralized forecast of BE allocations (3) to drive CAC decisions on incoming GP calls, we make use of a distributed team of DMs to solve PAPs I–III in a decentralized way. Let $\boldsymbol{m}_{h\kappa}(\cdot)$ be the information vector available for $DM_{h\kappa}$ with respect to the state of the overall network. Different measurable variables may be grouped in the $\boldsymbol{m}_{h\kappa}(\cdot)$ vector, depending on the PAP we deal with. We denote by $\hat{t}$ a reasonable upper bound for the time delay necessary for each DM to obtain stable information updates about the $\boldsymbol{m}_{h\kappa}(\cdot)$ vectors from the other DMs. Pricing reallocations are then performed by the DMs for each service class $h\kappa$ at consecutive time instants $t = k\hat{t}$, $k = 0, 1, \cdots, \infty$, on the basis of a "knowledge" collected as:

$$\boldsymbol{I}_{h\kappa}(k\hat{t}) = col\{\boldsymbol{m}_{h\kappa}((k-\Xi)\hat{t}), \cdots, \boldsymbol{m}_{h\kappa}((k-1)\hat{t})\} \quad (9)$$

where Ξ denotes the depth of such finite-horizon memory. The information vector of the DM constitutes a belief state of the real traffic condition of the network in the presence of time varying loads and users' responsiveness. When dealing with general traffic statistics or complicated formulations as for PAP II and III, the belief state cannot be derived in closed form by discrete-time equations as in [6]. The information vector is thus heuristically composed of different measurable quantities deriving from "*experienced insight into the specific system of interest*" [21]. The aim is to constitute the necessary *sufficient statistic* to properly infer price reallocations along time. More specifically, the variables of interest for PAP I are the number of GP requests received, for each service class, in the last time intervals of observation, together with the corresponding pricing assignments. Other quantities are relevant for PAP II and PAP III. Concerning PAP II, DMs' information vectors contain (together with the previous pricing assignments) the percentage of connections blocked, owing to users' refusals due to high prices. Concerning PAP III, DMs' information vectors are composed of quantities as for PAP I, together with the bandwidth available for the BE traffic and the corresponding BE revenue. The inference capability of these information vectors has been extensively validated by simulation analysis and, concerning PAP III, by results reported in [30]. Let us consider the PAP I stated in (6). Let $J_{k\hat{t}}(\boldsymbol{p}(k\hat{t})) = E_{\boldsymbol{\omega}} L_{k\hat{t}}[\boldsymbol{p}(k\hat{t}), \boldsymbol{\omega}]$ be the average-reward, infinite-horizon revenue functional after the price reallocation at time $k\hat{t}$:

$$L_{k\hat{t}}[\boldsymbol{p}(k\hat{t}), \boldsymbol{\omega}] = \lim_{T\to\infty} \frac{1}{T} \int_{k\hat{t}}^{k\hat{t}+T} \sum_{h=1}^{H} \sum_{\kappa=1}^{K} n_{h\kappa}(\tau) p_{h\kappa}(k\hat{t}) y_{h\kappa}(\tau)\, d\tau \quad (10)$$

and let $f_{h\kappa}(\boldsymbol{I}_{h\kappa}(k\hat{t}))$ be the *price reallocation law* of $DM_{h\kappa}$: $p_{h\kappa}(k\hat{t}) = f_{h\kappa}(\boldsymbol{I}_{h\kappa}(k\hat{t}))$. We denote by $\boldsymbol{f}(\cdot)$ and $\boldsymbol{I}(\cdot)$ all the DMs' reallocation laws and the related information vectors. The revenue functional (10) becomes the basis of the following functional optimization problem. **Problem functional - pricing allocation problem (F-PAP)**: Find the optimal pricing reallocation function $\boldsymbol{f}^*(\cdot)$, such that the following functional performance index is maximized:

$$J_{k\hat{t}}[\boldsymbol{f}(\boldsymbol{I}(k\hat{t}))] = E_{\boldsymbol{\omega}} L_{k\hat{t}}[\boldsymbol{f}(\boldsymbol{I}(k\hat{t}), \boldsymbol{\omega})] \quad (11)$$

The solution of F-PAP yields an "open-loop feedback" control law, able to perform on-line dynamic reactions to variable system conditions. Other F-PAPs can be obtained, in a similar way, by substituting to (10) the other revenue functions (7) and (8) related to PAP II and PAP III. The centralized formulation of F-PAP might appear in contrast with the decentralized structure of the DMs. Nevertheless, it is essential to obtain prices as functions of the overall state of the network. Actually, the price allocated by $DM_{h\kappa}$ influences not only class $h\kappa$ (and all the other service classes having some links in common with such a class), but also any other "remote" class, topologically far away from class $h\kappa$. The rationale of this complex relationship relies on the reciprocal influence of the classes sharing the bandwidth under the assigned network topology and routing scheme. As such, it is necessary that each DM maintain information about the overall state of the network. In this view, DMs periodically exchange the information related to the service classes they are monitoring, by means of dedicated signaling messages. After that, the decentralization of the price reallocation laws is obtained as follows. Once the optimal control law $\boldsymbol{f}^*(\cdot)$ solving (11) is obtained, it is memorized by each DM, thus allowing a virtual centralized computation of the prices. The process of replicating $\boldsymbol{f}^*(\cdot)$ in each DM is a natural consequence of the neural scheme we are going to investigate in the next section. It is finally worth noting that the infinite horizon functional (10) can be only computed through a simulation-based receding-horizon (RH) approximation; for simpler formulations than PAP I, the RH approach is used [6]. The neural approximating scheme we are going to investigate follows this direction, by exploiting a finite horizon estimate of (10) for each possible stationary configuration of the stochastic vector $\boldsymbol{\omega}$.

V. THE OPTIMIZATION METHODOLOGY

In order to approximate the optimal pricing control law $\boldsymbol{f}^*(\cdot)$, we develop a modified version of the *extended Ritz* method [31]. The extended Ritz method is a technique suitable for the approximation of the solution of functional optimization problems, by fixing the structure of the decision functions. Such decision functions are constrained to take on the structure of approximating networks, i.e., linear combinations of (nonlinear) basis functions, containing free parameters to be optimized:

$$\overline{\boldsymbol{f}}(\boldsymbol{I}, \boldsymbol{w}) = col\left\{ \sum_{l=1}^{\nu} c_l^i \zeta(\boldsymbol{I}, \tilde{\boldsymbol{w}}_l) + c_0^i;\ i = 1, \cdots, HK \right\} \quad (12)$$

where $\zeta(\cdot,\cdot)$ and ν represent a suitable basis function and the number of basis functions used to build the approximator $\overline{\boldsymbol{f}}(\cdot,\cdot)$, respectively. The extended Ritz usually needs closed-form formulas for the differential equations of the controlled system and for the cost functional. In this work, we adapt it to a *discrete event simulation environment*, in which system dynamic equations are not required explicitly and only estimates of the cost functional and its gradient can be obtained. In this perspective, the proposed technique generalizes the results of [31], [32], in which only stationary stochastic environments (involving optimization problems in different research fields, such as packet-based routing, freeway traffic control, systems' dynamic parameters estimation, target motion analysis) are considered for performance evaluation. Among the possible choices of structures of the form (12), we choose *one hidden layer feed-forward neural networks* (due to their powerful approximation properties to face the possible exponential growth in the number of free parameters, needed to obtain an increasing degree of accuracy). Then, we have:

$$\overline{\boldsymbol{f}}(\boldsymbol{I},\boldsymbol{w}) = col\left\{\sum_{l=1}^{\nu} c_l^i \sigma(\tilde{\boldsymbol{w}}_{1l}^T \boldsymbol{I} + \tilde{w}_{0l}) + c_0^i;\ i = 1,\cdots,HK\right\} \quad (13)$$

where $\tilde{\boldsymbol{w}}_l = col\{\tilde{\boldsymbol{w}}_{1l}, \tilde{w}_{0l}\}, l = 1,\cdots,\nu$, and $\sigma(\vartheta)$ is a nonlinear activation function, e.g., a *sigmoidal* ($\sigma(\vartheta) = \frac{1}{1+e^{-\vartheta}}$) or a *hyperbolic tangent* ($\sigma(\vartheta) = \frac{e^{\vartheta}-e^{-\vartheta}}{e^{\vartheta}+e^{-\vartheta}}$) function. We suppose that each price $p_{h\kappa}$ is constrained to a given domain, i.e., $p_{h\kappa} \in \left[p^{m_{h\kappa}}; p^{M_{h\kappa}}\right]$. In order to guarantee the fulfilment of such constraints, we compose the output of the neural network with a *normalization operator* $\Theta(\cdot)$. We thus obtain prices' reallocations $\boldsymbol{p}(k\hat{t})$ at any time $k\hat{t}$ as:

$$\begin{aligned}
&\boldsymbol{p}(k\hat{t}) = \Theta(\overline{\boldsymbol{f}}(\boldsymbol{I}(k\hat{t}),\boldsymbol{w}))\\
&p_{h\kappa}(\overline{\theta}_{h\kappa}) = p^{m_{h\kappa}} + (p^{M_{h\kappa}} - p^{m_{h\kappa}})\overline{\theta}_{h\kappa};\\
&\overline{\theta}_{h\kappa} = \overline{f}_{h\kappa}((\boldsymbol{I}(k\hat{t}),\boldsymbol{w})),\ \overline{\theta}_{h\kappa} \in [0.0, 1.0],\ \forall h,\kappa
\end{aligned} \quad (14)$$

We shall call "neural pricing allocation function" (NPAF) the aggregation of functions (14), obtained as composition of the neural networks (13) and the normalization operators, and denote it by $\hat{f}((\boldsymbol{I}(\cdot),\boldsymbol{w}))$. It follows that a parametrized revenue function is obtained, by substituting the fixed structure of such NPAF into (11), depending on the parameter vector $\boldsymbol{w}$, thus leading to the following mathematical programming problem.

Problem F-PAP$_w$: Find the optimal parameter vector $\boldsymbol{w}^*$, such that the revenue function (15) is maximized:

$$E_{\boldsymbol{\omega}} L_{k\hat{t}}\left[\hat{\boldsymbol{f}}(\boldsymbol{I}(k\hat{t},\boldsymbol{w}),\boldsymbol{\omega})\right] = E_{\boldsymbol{\omega}} \tilde{L}_{k\hat{t}}(\boldsymbol{w},\boldsymbol{\omega}) \quad (15)$$

In this way, the F-PAP (11) has been reduced to an unconstrained nonlinear programming problem. To solve such nonlinear programming problem, we should apply a gradient-based algorithm:

$$\boldsymbol{w}^{\chi+1} = \boldsymbol{w}^{\chi} - \xi \nabla_{\boldsymbol{w}} E_{\boldsymbol{\omega}} \tilde{L}_{k\hat{t}}(\boldsymbol{w}^{\chi},\boldsymbol{\omega}), \chi = 0,1,2,\cdots,\infty \quad (16)$$

where ξ is a fixed step-size. However, the explicit computation of the expected revenue and its gradient is a very hard task, even if closed-form formulas for the functional $\tilde{L}_{k\hat{t}}(\cdot)$ were available [31]. We choose to compute a realization $\tilde{L}_{k\hat{t}}(\boldsymbol{w}^{\chi},\boldsymbol{\omega}^{\chi})$, instead of the gradient $\nabla_{\boldsymbol{w}} E_{\boldsymbol{\omega}} \tilde{L}_{k\hat{t}}(\boldsymbol{w}^{\chi},\boldsymbol{\omega})$, and we apply the updating algorithm:

$$\boldsymbol{w}^{\chi+1} = \boldsymbol{w}^{\chi} - \xi_{\chi} \nabla_{\boldsymbol{w}} \tilde{L}_{k\hat{t}}(\boldsymbol{w}^{\chi},\boldsymbol{\omega}^{\chi}),\ \chi = 0,1,2,\cdots,\infty \quad (17)$$

where the index χ denotes both the steps of the iterative procedure and the generation of the χth realization of the stochastic processes involved in $\boldsymbol{\omega}$. The components of the gradient $\nabla_{\boldsymbol{w}} \tilde{L}_{k\hat{t}}(\boldsymbol{w}^{\chi},\boldsymbol{\omega}^{\chi})$ can be obtained by using the classical backpropagation equations for the training of neural networks. The backpropagation procedure must be initialized by means of the quantities $\partial L/\partial p_{h\kappa}$, $h = 1,\cdots,K$; $\kappa = 1,\cdots,K$ (i.e., the gradient $\nabla_{\boldsymbol{p}} L(\boldsymbol{p},\boldsymbol{\omega}^{\chi})$). Unfortunately, in our case, such quantities cannot be obtained analytically as in [31], because no closed-form of the functional $L(\cdot)$ is available [6]. Hence, during the training phase (17), gradient estimates are computed as:

$$\frac{\partial L_{k\hat{t}}(\boldsymbol{p}(k\hat{t}),\boldsymbol{\omega}^{\chi})}{\partial p_{h\kappa}(k\hat{t})} \cong \frac{L_{k\hat{t}}(\boldsymbol{p}_{h\kappa+}(k\hat{t}),\boldsymbol{\omega}^{\chi}) - L_{k\hat{t}}(\boldsymbol{p}(k\hat{t}),\boldsymbol{\omega}^{\chi})}{\Delta_p} \quad (18)$$

where $\boldsymbol{p}_{h\kappa+}(k\hat{t})$ means that a "small" increase is carried out for $p_{h\kappa}(k\hat{t})$ (the $h\kappa$th component of the price vector $\boldsymbol{p}(k\hat{t})$), i.e., $p_{h\kappa+}(k\hat{t}) = p_{h\kappa}(k\hat{t}) + \Delta_p$. Since such training procedure can be performed off line, $L_{k\hat{t}}(\boldsymbol{p}_{h\kappa+}(k\hat{t}),\boldsymbol{\omega}^{\chi})$ and $L_{k\hat{t}}(\boldsymbol{p}(k\hat{t}),\boldsymbol{\omega}^{\chi})$ are computed by simulation inspection through a suitable *discrete event simulator* that mimics the behaviour of the network.

A. Remarks on the Neural Pricing Allocation Function and Its Application in Real Time

i) *Computational effort*: The approach deriving from (17) and (18) recalls to some extent the gradient-based procedure developed in [6] for a MDP applied to a slightly modified version of our PAP I. The difference is that the above-described training procedure can be performed off line. The related computational burden does not influence the on-line performance of the system. In real time, the optimized NPAF $\hat{\boldsymbol{f}}(\cdot,\boldsymbol{w}^*)$ is applied, obtaining new price reallocations "almost instantly". Once a neural network is trained, the computation of its input-output relationship requires a given number of multiplications depending on the number of neural units. Such a number is, in the worst case, super linear (and never exponential) in the number of inputs [31]. Although special computation for the (e.g., sigmoid) neural units is required, the overall computational complexity is very limited (see e.g., [33] for details). The solution of PAP III includes a sort of decentralized BE revenue forecast (similar to the one of [30]) to drive incentive or discouragement to GP calls. Without the neural approximation of (15), such forecast would require the solution of problem (3), whose computational time is exponential in the number of users (we verified this trend by extensive simulation campaigns [12], [30]). The advantage is significant with respect to other computationally expensive algorithms, developed for GP pricing [4], [5], [7], [8], [14], [15], too.

ii) *Non-stationarity*: Our approach can accommodate non-stationarity. For the F-PAPs investigated here, an optimal pricing assignment arises for each possible combination of stationary traffic laws $\boldsymbol{\lambda}(\boldsymbol{p})$ (PAP I) and utility functions $\boldsymbol{U}(\cdot)$ (PAP II and PAP III), thus guaranteeing the maximization of the long-term expected revenue performance. Variable system conditions are related to non-stationary processes coming from time-varying $\boldsymbol{\lambda}(t,\boldsymbol{p})$, $\boldsymbol{U}(t,\cdot)$. The information vector (9) "captures" the current structure of the stochastic environment and the optimized NPAF automatically tunes the prices, without any precise insight into the structure of the current stochastic vector $\boldsymbol{\omega}$. Let Ω be the set of the possible compositions of the stochastic environment, i.e., $\Omega = \{\boldsymbol{\omega}_1,\cdots,\boldsymbol{\omega}_{|\Omega|}\}$. The inference capability of the NPAF needs a training phase with respect to the stochastic set Ω. This means that the SP, during the planning phase of the network, defines the possible operating behaviours of the users (on the basis of its experience and of measurements performed on the network in the past), thus enabling the set Ω to aggregate the possible stochastic environments, defining the future network evolution. Then, the training phase (17) is performed, to obtain the optimized NPAF. The SP also periodically supervises the users' behaviour in real time, in order to exploit other possible instances of Ω, not previously used in the training phase, and restarts a new training to update the NPAF, if necessary. This updating procedure is briefly summarized in the Appendix. The key point is that the training applied in real time is faster than the one performed for the first time off line, because an "already partially trained" NPAF is adopted [33].

iii) *Distributed DMs and scalability*: DMs may be located within the applications servers responsible for the on-line monitoring and control of the charging system (see e.g., Chapter 3.11 of [28]). A dedicated signalling is required for them to exchange information about the service classes. The QoS-based variant of BGP [34], defining a common language for network entities to exchange generic policy information, is well suited for our decentralized paradigm. In this view, the implicated communication scheme recalls the BGP protocol establishment of interior BGP (iBGP) sessions within an autonomous system (AS). iBGP peers must be fully meshed together, as in our case. For a large AS, say in the order of several hundred iBGP sessions, it becomes impractical to manage such a large number of sessions. To avoid iBGP session mesh and yet still be able to propagate information to all BGP routers inside the AS, the route reflector (RR) concept is used. The RR is responsible for replicating messages towards its peers, thus avoiding a mesh structure of the peers. The same idea can be applied to our DMs in order to limit scalability problems. The time scale of the DMs signalling depends on the parameter $\hat{t}$, which defines the time granularity of the updates made on the DMs' information vectors. It has an upper bound, which depends on the size of the period of traffic stationarity T_s. To let reallocations work properly, the NPAF must include in T_s at least one reallocation step comprising the entire depth of its information vector. According to (9), this implies $\hat{t} \leq T_s/n_s\Xi$, $n_s \in \mathbb{N}^+$, $n_s \geq 2$ being the number of reallocation steps one wants to include within T_s. Fig. 1 may help understand; it considers the worst case for $\hat{t}$ ($n_s = 1$), where the information vector acquired over a given stationary horizon (e.g., "VoIP environment") drives the reallocation over a new stationary horizon ("VoIP & video environment"). The parameter Ξ represents the depth (i.e., number of samples) of the information vector, at each reallocation step, necessary to capture accurate estimation of traffic statistics. Setting large values of Ξ means increasing both estimation accuracy and communication overhead (and viceversa). General rules are not available in these cases, because the optimal setting depends on both the stochastic environment and the revenue functional. Heuristically setting Ξ is the only solution. In the scenarios considered below, we found $\Xi = 5$ to produce the best compromise between accuracy and communication overhead. Simulation results, not reported in the paper, corroborate this value for a wide range of topologies and traffic laws. The order of magnitude of T_s, on the other hand, deserves a more insightful analysis. In case it could be measured in minutes, the effect on $\hat{t}$ would lead to an impractical signaling overhead. However, as outlined in [14], demand patterns in the Internet suggest that user sensitivity is piecewise stationary during the time horizon of the day (e.g., with 3 or 4 time intervals of stationarity). It is therefore reasonable to assume that the time granularity of the measures collected on the information vectors ($\hat{t}$) can be easily set to much lower values than the time scale of traffic stationarity (T_s), without incurring in negative consequence in terms of communication overhead or loss of important information. An example concerning the coexisting effect of both $\hat{t}$ and the number of DMs is reported in Subection VI.B.

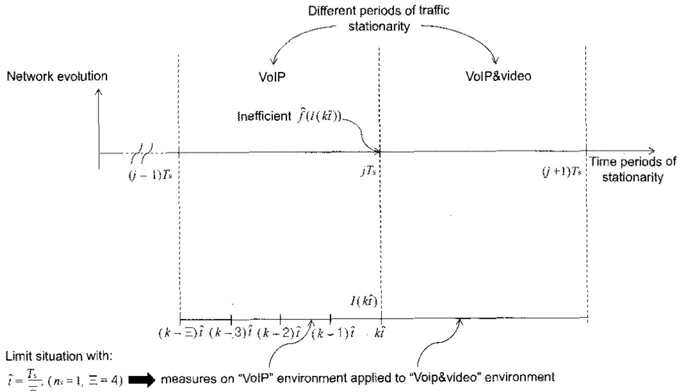


Fig. 1. Upper bound of DMs communication delay $\hat{t}$: Worst case.

VI. PERFORMANCE EVALUATION AND DISCUSSION

In this section we illustrate an evaluation of our pricing control mechanism. To this aim, we have developed a simulation tool in C++ that describes the behavior of the network at the flow level for both GP and BE traffic. In the first part of our performance evaluation, we are interested in analyzing the convergence behavior of the training algorithm (17), by employing the gradient estimation procedure (18) and assuming stationary conditions. The aim is to emphasize, through simulation inspection, that the proposed neural approach yields the optimal pricing assignment after the training phase. Then, we highlight how variable system conditions (in this case: $\boldsymbol{\lambda}(t, \boldsymbol{p})$, $\boldsymbol{U}(t, \boldsymbol{y})$) can be mapped on the neural approximator, and we compare its online performance with an optimization technique that exploits a perfect knowledge of the users' utility functions.

A. Convergence Behaviour of the Control Algorithm

The first simulation scenario is related to PAP I. It is aimed at introducing the parameters of the following simulation scenarios and at detailing both the implementation of the training phase and the procedure performed to verify the optimality of the NPAF. In this case, a trivial optimal pricing assignment (easily computable through simulation comparisons) has to be reached for a small network (composed by one link and two service classes). As outlined above, a service class is defined as a routing path combined with a specific QoS treatment (in terms of equivalent bandwidth assignment). The simulation scenario is built as follows. Two routes, whose traffic demands are $\lambda_i(p_i) = 1.0p_i^{-1.05}$; $p_i \in [1; 100]$; $i = 1, 2$, belong to a single link of 10.0 Mbps capacity. The corresponding interarrival times are exponentially distributed. The aforementioned traffic laws are taken from [35] and reproduce the demand elasticity of a voice service. Each voice source is modelled as an exponentially modulated on-off process, with the mean on and off periods being of 1.008 s and 1.587 s duration, respectively, as per the ITU P.59 recommendation. The source peak rate is 16.0 kbps. Bandwidth requirements follow an equivalent bandwidth expression, derived by simulations, to assure a *packet loss* of 2%. The buffer dimension is fixed at 100 voice packets (80 bytes each) to guarantee a maximum end-to-end delay lower than 150 ms. The mean duration of the calls is fixed to 10.0 minutes and is log-normally distributed. The training phase acts as follows. At the χth step of the training algorithm (17), a sample path $\boldsymbol{\omega}^{\chi}$ of the stochastic variables $\boldsymbol{\omega}$ is generated on a time horizon $[0, \Xi\hat{t} + T]$, accordingly to the underlying probability distributions. At time $\Xi\hat{t}$, once that the information vector $\boldsymbol{I}(\Xi\hat{t})$ is collected, the neural price reallocation function is applied and the corresponding performance is computed, together with the derivative estimates (18) for each price component. Then, the backpropagation procedure is applied to train the NPAF and a new step of (17) is performed, until a steady state of the revenue function (6) has been reached. A neural network with 15 hyperbolic tangent neural units in the hidden layer and with a sigmoidal output layer has been used. The architecture of this approximator was determined experimentally by progressively increasing its complexity, until no significant increase in the expected revenue function occurred. A new interval of observation starts every hour ($\hat{t}$=1 hour) and $\Xi\hat{t}$ is fixed to 6 hours (the first hour of simulation during training is considered a transient period to meet the regime of the stochastic processes). The overall simulation time T in (6) after the price reallocation is set to 15 hours for each training step ($\Xi\hat{t} + T$=15+6 hours). The gradient stepsize ξ_χ of (17) is taken as $\xi_\chi = \frac{1.0}{2.0 \cdot 10^5 + \chi}$ (thus assuring a decreasing behaviour as convergence requirement); we have also added a "momentum" term $\rho(\boldsymbol{w}^{\chi} - \boldsymbol{w}^{\chi-1})$, ρ=0.3, to (17), as usually done in training neural networks to speed up convergence. The Δ_p parameter of the gradient estimation procedure (18) was fixed to 6.0. We intentionally fixed the training parameters in order to obtain 100 training steps, just to highlight the convergence behaviour. Figs. 2 and 3 show the revenue performance and the prices' assignments during the training phase. For the simple network scenario under investigation, the lowest price values p_i^o=1.0, $i = 1, 2$ (and the corresponding frequency of the interarrival requests $\lambda_i^o = \lambda_i(p_i^o)$) guarantee the best performance. As we are going to illustrate, this trivial situation does not always hold true in more general traffic conditions and with more complicated routing assignments. For the time being, it is worth noting that the prices' optimality (reached after training) can be confirmed by simulation inspection. Namely, we employed a gradient-based descent, directly over the prices' domain, i.e., $\boldsymbol{p}^{\varphi+1} = \boldsymbol{p}^{\varphi} - \psi \nabla_{\boldsymbol{p}} L_{k\hat{t}}(\boldsymbol{p}^{\varphi}, \boldsymbol{\omega}^{\varphi})$, $\varphi = 0, 1, 2, \cdots$ (the gradient is again estimated as in (18)) and found that the optimal point (achieved at the end of the gradient procedure) is the same as the one obtained after training. We also obtained the same optimum by using Powell's algorithm, which is a well-known non-linear programming technique that does not exploit any knowledge on the gradient of the performance index.

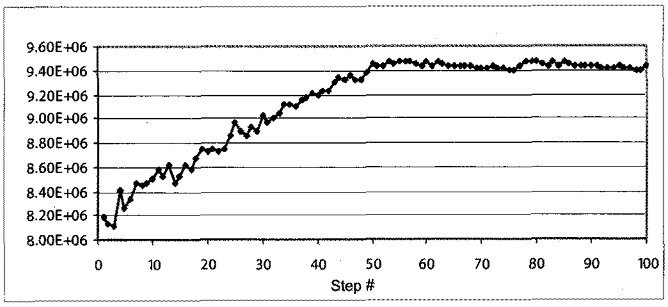


Fig. 2. Revenue during training.

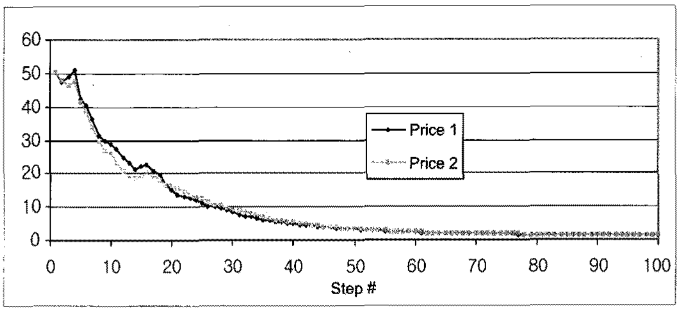


Fig. 3. Prices during training.

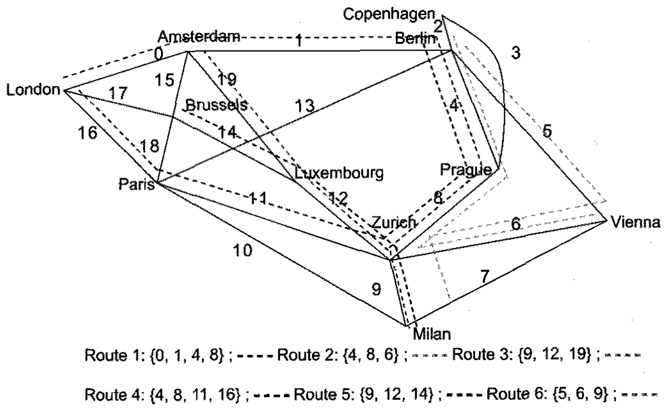


Fig. 4. Topology of the test network.

B. Optimal Pricing Assignment in a Real Network Scenario

The second simulation experiment regards a "VoIP" scenario with PAP I and makes use of the COST 239 experimental network topology under the voice service classes described above and deployed along the routes depicted in Fig. 4. The capacity of the network links is fixed to 30.0 Mbps. The employed neural network structure is the same as the one of the previous simulation scenario. This time, due to the combinatorial number of the possible trials in the pricing assignment, it is more difficult to find out the optimal solution through simulation inspection. The prices' domain, the training parameters, the simulation horizon T in (6), and the Δ_p parameter were accurately tuned in order to speed up the convergence of the training phase: $p_i \in [1, 10]; \forall i$, $\xi_\chi = \frac{1.0}{6.0 \cdot 10^4 + \chi}$, $\rho = 0.3$, T=1000 hours, Δ_p=1. The training procedure (after 10 steps, around 7 minutes on an AMD Athlon @1.8 GHz) converges to the following pricing assignment:

Pricing assignment	Revenue	Blocking probability
p_i = 100.0, i=1,...,6	4.406673e+006	0.0
p_i = 25.0, i=1,...,6	4.647619e+006	0.0
p_i = 10.0, i=1,...,6	4.863280e+006	0.0
p_i = 5.0, i=1,...,6	5.054878e+006	0.001132
(19)	5.125e+006	0.009606
p_i = 4.0, i=1,...,6	5.087591e+006	0.006751
p_i = 3.0, i=1,...,6	4.982476e+006	0.031672
p_i = 2.0, i=1,...,6	4.426812e+006	0.136192
p_i = 1.0, i=1,...,6	2.889035e+006	0.402790

Fig. 5. Pricing assignments and corresponding performance.

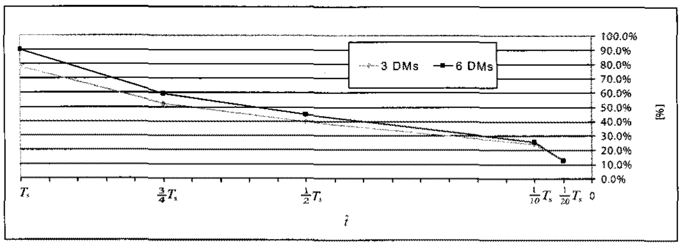


Fig. 6. Percentage decrease with respect to the PK strategy.

$$p_1^o = 5.346664; p_2^o = 5.647137; p_3^o = 3.616069;$$

$$p_4^o = 3.249087; p_5^o = 1.789551; p_6^o = 2.213421. \quad (19)$$

As mentioned above, we verified the optimality of (19) through a gradient-based procedure and Powell's algorithm, both performed over the price variables. We report in Fig. 5 the most significant samples obtained over the performance index, just to highlight how solution (19) guarantees the best bandwidth sharing among the users, corresponding to an overall blocking probability of 0.9606%, which reveals to be the optimal one (in terms of revenue), as compared to the other blocking probability values (obtained in correspondence with different price allocations). As far as non-stationary conditions are concerned, we considered the addition of the demand function $\lambda(p) = 2.0p^{-2.50}$ in substitution (for the routes 3, 5, 6) of the voice services adopted above. This situation is called "VoIP & Video" scenario. The function $\lambda(p) = 2.0p^{-2.50}$ is related to video services [35]. The mean duration of the calls is 10.0 minutes and is exponentially distributed. Taking [8] as a reference, each video source is modelled as an exponentially modulated on-off process, with mean rate 0.5 Mbps and peak rate 3.0 Mbps. The average burst is 10.0 s. Video bandwidth requirements follow an equivalent bandwidth expression (found by simulation) to assure a Packet Loss of 10^{-5} (taking the ATM cell structure as a reference). Lower loss probabilities can be taken into account by using a closed-form formula of the equivalent bandwidth. The buffer dimension for the video service is fixed to 30 ATM cells for each network link, to guarantee a maximum delay per node always lower than 20 ms. The optimized NPAF must optimally tune the prices in dependence of the changes in the stochastic environment (in the periods of time when the video service is multiplexed with the voice one). Using the updating algorithm summarized in the Appendix, a new training phase is obtained (with a computational time of about 4 minutes to complete on the same desktop PC) with respect to the new traffic demand (the video service). The updated NPAF "learns" that the prices' allocation in the presence of video services is slightly different from (19), assigning $p_3^o = 1.130131$; $p_5^o = 1.924691$; $p_6^o = 1.411997$ (thus providing incentives for the video calls, since they reveal to be more lucrative for the SP). Fig. 6 highlights the performance obtained by a distributed team of DMs, equipped with the trained NPAF (as outlined in Section IV), with respect to an unfeasible "perfect knowledge" (PK) strategy that exactly knows the time changes of a sequence of consecutive 10 VoIP and VoIP & video scenarios (one scenario is repeated after the other every T_s=5 hours). The width of the confidence interval over the following simulated revenue performance measures is less than 1% for 95% of the cases. The PK strategy controls the prices accordingly to the neural solutions above with respect to each different scenario. The rationale of the performance decrease envisaged in Fig. 6 relies on the imperfect allocations made by the DMs in correspondence of the stationarity changes; it is an increasing function of $\hat{t}$ (the time granularity of the DMs' information vectors). Also the number of DMs has an impact on performance: When only 3 DMs are used, the performance decrease is more limited (one DM is assigned to each couple of routes: Routes 2 & 6 with link 6 in common, DM in Zurich; routes 3 & 5 with link 12 in common, DM in Milan; routes 1 & 4 with link 8 in common, DM in Prague). When 6 DMs are employed, a DM is assigned to each different traffic route. This effect is due to the slow propagation of the information acquired on the overall state of the network as the number of DMs increases. However, a reasonable value for the updating speed of the information vectors results to be $\hat{t} \leq T_s/10 = 30$ min (it guarantees no more than 20% of performance decrease). As such, the communication overhead required by the team of DMs is almost irrelevant.

C. *Dynamic Reactions to Variable Utility Functions*

After evaluating the optimality guaranteed at the end of the training phase, we now test the NPAF's on-line performance, by taking into account an instance of PAP II. We introduce variable GP users' utility functions, i.e.: $U(t, y)$. In a network composed by a single link of 10.0 Mbps capacity, two service classes, whose frequency demands are both 3.0 calls per minute (exponentially distributed), are introduced, together with utility functions of the form:

$$U_i(y_s) = \alpha_i \sqrt{y_s}; i = 1, 2; \; y_s \in [0.1, 1.0]. \quad (20)$$

The index i denotes the class of service and α_i is a random variable, associated to each incoming connection, uniformly distributed in the intervals [0.0, 60.0] or [0.0, 80.0] for each class, in different time periods. This means that the average willingness to pay of one class is periodically sensibly higher than the one of the other class. The NPAF should be able to discover these statistical changes and optimally set the prices assigned to each class. During the network evolution, each user s requires a service rate y_s (uniformly distributed in the range [0.1, 1.0]

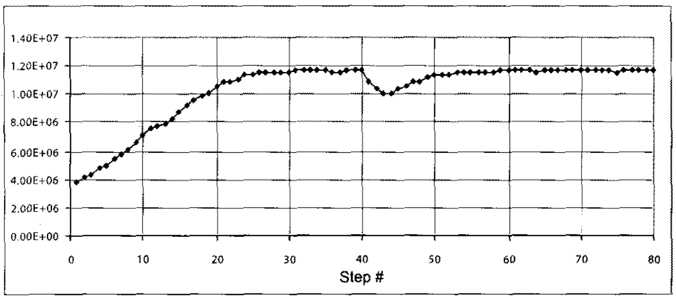


Fig. 7. Revenue during training.

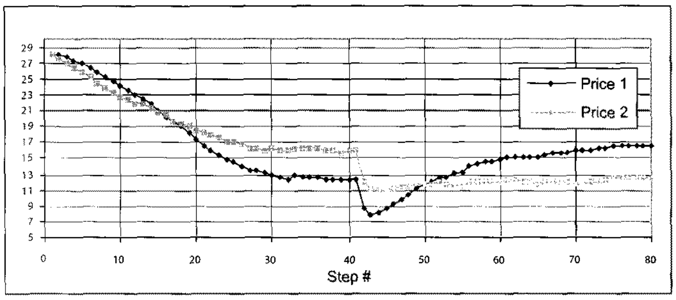


Fig. 8. Prices during training.

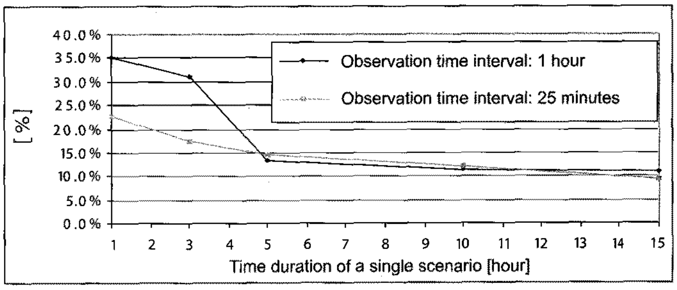


Fig. 9. Percentage decrease with respect to the PK strategy.

Mbps) and, on the basis of the imposed price p_s, accepts to enter the network (and pay the corresponding tariff $p_s y_s$, e.g., in \$ per minute) on the basis of its willingness to pay ν_s, i.e., if $p_s \geq \nu_s = y_s U_s^{'}(y_s)$. Each time a new user arrives, its willingness to pay is randomly assigned through (20). The mean durations of the calls are exponentially distributed, with an average of 10.0 minutes. The prices' domain, the employed neural network structure and the training parameters are equal to the ones of the first simulation scenario. The previous 5 time intervals of observation (of one hour duration each) are taken into account in the information vector. A preliminary training phase is shown, consisting of 80 steps, to highlight the neural price function reactions to variable system conditions. The first 40 steps are related to utility functions taken from (20), whose α_i parameters are uniformly distributed between [0.0, 60.0] and [0.0, 80.0] for the first and the second class, respectively (then, the second class guarantees a higher willingness to pay). In the last 40 training steps, the situation is reversed and the first class shows a higher willingness to pay, on average. The revenue performance and the prices during training are depicted in Figs. 7 and 8, respectively. At the end of the first 40 steps, the training procedure has discovered the optimal prices' assignment (i.e., $p_1^o = 12.7$ and $p_2^o = 16.0$) with respect to the first "average" utility function scenario. On the other hand, when the second utility function scenario is introduced (after the first 40 steps), a revenue decrease arises and 20 additional training steps are necessary to find out the new optimal prices' assignment, corresponding to $p_1^o = 16.8$ and $p_2^o = 12.9$. Up to 320 independent repetitions of such training phase are necessary to map the variable system conditions (i.e., $\boldsymbol{U}(t, \boldsymbol{y})$) on the NPAF. In Fig. 9, we depict the on-line performance of the trained NPAF. The width of the confidence interval over the following simulated revenue performance measures is less than 1% for 95% of the cases. The test consists in alternating the aforementioned average utility function scenarios at different time intervals after the training phase. Our approach is compared with an infeasible PK strategy that exploits a perfect knowledge of the users' utility functions and changes the prices accordingly to the results shown in Fig. 8. Namely, when the first service class has more willingness to pay, the PK strategy sets the prices as $p_1^o = 12.7$ and $p_2^o = 16.0$, and vice-versa when the second traffic class gives rise to "higher" utility functions. We recall that the updating parameters of the information vector (Section V.A.iii)) are: $\hat{t} = 1$ hour, $n_s = 2$ and $\Xi = 2$. Looking at the "*observation time interval: 1 hour*" curve in Fig. 9, it is easily observable that when the two static scenarios alternate every 5 hours, the percentage decrease in terms of revenue over the PK strategy is below 15%. On the other hand, when they alternate more frequently, the revenue decrease is much more evident (up to 35% when the duration of each static scenario is one hour). This means that when the duration of the "static" scenarios is significantly smaller than the one of the observation horizon, the neural network does not infer correctly the next prices' reallocation. Such a drawback can be faced by properly tuning the sample time intervals of the information vector. If the training phase is repeated with observation time intervals of 25 minutes each, the trained NPAF appreciably improves the performance ("*observation time interval: 25 minutes*" curve in Fig. 9).

D. Multiplexing GP and BE Traffic

We next turn our attention to the PAP III, thus analyzing GP and BE traffic mixed together. In this test, we take the second simulation scenario (with the voice demand function concerning GP traffic) as a reference and we introduce BE flows in the routes 1, 2, and 4 in Fig. 4. The parameters of BE traffic are fixed as follows. The mean interarrival time of the BE connection requests is 1.0 call per minute (exponentially distributed) and the mean distribution of a call is 10 minutes (log normally distributed). The utility functions are:

$$U_r(x_r) = \sqrt{2}\tau^{-1}\sigma_r \arctan\left(\frac{1}{\sqrt{2}}\tau x_r\right); \; x_r \in [0.1, 1.0] \; \forall r \tag{21}$$

and are taken from [19] to mimic the TCP *Reno* behaviour. τ denotes the *round trip time* of each TCP connection. For the sake of simplicity, we suppose that the optimal equilibrium point (found at the end of the training phase) assures an amount of bandwidth for the BE traffic sufficient to disregard the packet-based congestion in the link's BE queues. Hence, τ can be fixed

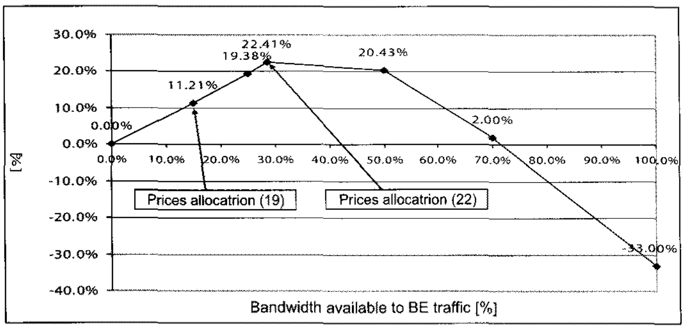


Fig. 10. Improvement with BE traffic: 1.0 BE call per minute.

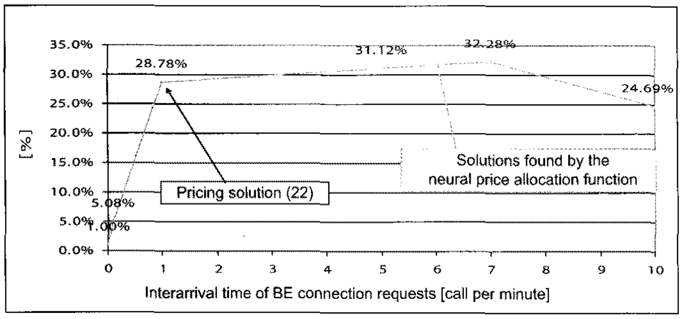


Fig. 11. Improvement with BE traffic: Variable BE requests.

only on the basis of the propagation delay of the links (supposed to be equal to 10 ms). Then, τ= 80 ms for routes 1 and 4; τ= 60 ms for route 2. σ_r is a random variable (uniformly distributed in [10, 20]), acting as a scaling factor of the willingness to pay of the BE users. A non-linear programming solver [36] is integrated in the simulator, to mimic the behavior of the BE portion of the network. Each time a change in the bandwidth available for the BE traffic is detected (i.e., when a GP call enters or exits the network) the welfare problem (3) is solved to compute the new equilibrium point in the BE bandwidth allocation; then, the new BE prices are calculated through (1) with respect to such new equilibrium. In this way, we take into account the variability of BE prices due to the bandwidth sharing at the flow level as in (4) and (5). The BE prices are considered always at equilibrium, disregarding their transient fluctuations before reaching a new operating point. Modelling the behaviour of the BE portion through this mechanism is suitable for the off-line training of the NPAF. It allows accelerating the training phase, since it does not require a packet-based granularity of the simulations. The NPAF's structure and the training parameters are equal to the ones of the first simulation scenario. Fig. 10 highlights the percentage improvement over the optimum revenue accomplished by (19) (in the second simulation scenario), as a function of the percentage of bandwidth available for the BE traffic. This latter index is defined as the value of the bandwidth available in the bottleneck links of BE routes (links 4 and 8), averaged along the time horizon T of a training step. The new optimal solution is:

$$p_1^o = 6.414321; p_2^o = 5.411827; p_3^o = 8.913673;$$

$$p_4^o = 11.560122; p_5^o = 12.468861; p_6^o = 4.385691; \quad (22)$$

corresponding to 28.78% of bandwidth available for the BE traffic, which corresponds to the maximum of the curve depicted in Fig. 10. The optimal operating point achieves 22.41% of percentage revenue improvement when, on average, 28.78% of the BE bottleneck links' capacity is left to the BE traffic. On the other hand, the optimal solution (19), which ignores the presence of BE, would leave the BE traffic only 15% of the links' capacity. Fig. 10 was obtained throughout the training phase as regards the x-axis range [15%, 28.78%]. The remainder of the curve has been found out by simulation inspection, for the sake of completeness. At the end of the training phase, the NPAF found the new optimal solution (22), whose components 3, 4, 5 are higher than the ones obtained in (19). This means that less GP calls are required in the routes 3, 4, 5 to maximize the revenue performance, since the willingness to pay of BE users reveals to be more profitable when 28.78% of the bandwidth is available for the BE traffic. Fig. 11 shows the optimum operating point of the system, as a function of the frequency of the BE interarrival requests. The point [1.0, 28.78] corresponds to the aforementioned solution (22). The other points are obtained by repeating the training phase with different values of λ_{BE}. It is worth noting that similar results are obtained when another model of BE utility is used (i.e., TCP *Vegas*, utility of the form $U_r(x_r) = \alpha_r \tau_r \log x_r$ [19]). The optimal operating point with the TCP Vegas utility is [27.91%, 25.13%] instead of [22.41%, 28.78%]. This would suggest that, for the simulation scenario investigated here, changing the TCP model does not significantly affect the operating points achieved by the NPAF after training. The aggregate effect of BE users has a higher impact on the revenue performance, despite the changes in the BE utility functions. This latter result is quite surprising and suggests further experiments through simulations at the packet level (thus capturing the transient behaviour of the BE prices, too). This activity is currently the subject of ongoing research.

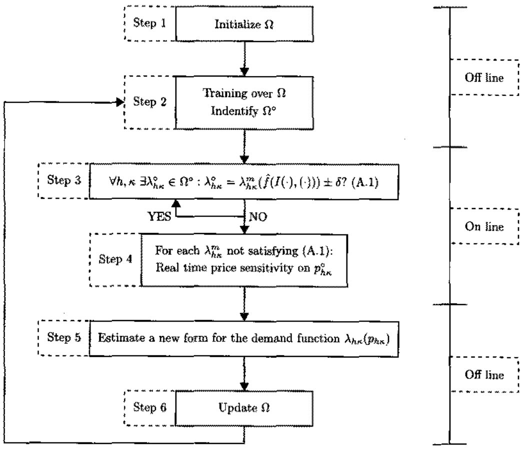
VII. CONCLUSIONS AND FUTURE WORK

In this paper, a novel control mechanism has been studied to allow optimal price reallocations as a function of the state of the network. Simulation results have shown how the *service provider*'s revenue is maximized, despite the possible changes in the traffic demands and in the willingness to pay of the users. Future work might include the application of further constraints, such as respecting fixed blocking probabilities for the GP service classes. The latter task can be accomplished by implementing proper penalty functions during the training phase (see e.g., [32]). We are also working on the evaluation of the NPAF's performance when it is employed in a decentralized fashion (as outlined at the end of Section V). The aim is to highlight the impact of decentralized control for GP pricing, which is currently an unexplored area of research.

APPENDIX

A. Update the NPAF in Real Time

Without loss of generality, we consider on-line variations of the stochastic vector as regards the demand functions $\boldsymbol{\lambda}(\cdot)$. The flow chart and a brief description of the algorithm is depicted



(a)

Step 1) The set Ω is initialized during the planning phase with respect to the predictable users' demand functions (possible instances of the vector ω).

Step 2) A first training phase is applied to find out the optimized NPAF (NPAF*) over Ω. Let Ω° be the set collecting the optimal operating points of the system with respect to the solutions of the F-PAP$_w$ (15) for every instance of the vector ω.

Step 3) In real time, it is easily recognizable if the NPAF* guarantees the optimal operating point. Let us define as $\lambda^m_{h\kappa}$ the measured frequency of the connection requests for the service class $h\kappa$. Using the information collected in the set Ω, we know that the optimal operating point $\lambda^m_{h\kappa}$ for service class $h\kappa$ is $\lambda^\circ_{h\kappa} = \lambda_{h\kappa}(p^\circ_{h\kappa})$, $p^\circ_{h\kappa} = \hat{f}(I(\cdot), \cdot)$. If $\lambda^m_{h\kappa}(\hat{f}(I(\cdot), \cdot))$ is far away from the ideal value $\lambda^\circ_{h\kappa}$, we therefore infer that a new user sensitivity for service class $h\kappa$ has taken effect.

Step 4) Then, we collect some points of the function $\lambda_{h\kappa}(p_{h\kappa})$ by varying the allocated price $p_{h\kappa}$. This operation is performed in real time along an estimation time horizon denoted by T_e (the number of samples acquired on the functions $\lambda^m_{h\kappa}(\cdot)$ is denoted by n^m). During this period of time, the allocated prices are suboptimal and are aimed at obtaining an on-line sensitivity of users' behaviour.

Step 5) On the basis of the collected measures at Step 4), we estimate a new form for the demand function $\lambda_{h\kappa}(p_{h\kappa})$. To do this, a simple regression algorithm, solved by a non-linear programming solver (e.g., [36]) is sufficient.

Step 6) We then update the set Ω with the new function $\lambda_{h\kappa}(p_{h\kappa})$ and restart the training phase over the updated set Ω and return to verify the performance of the NPAF in real time (Step 3)).

(b)

Fig. 12. Algorithm for updating the NPAF in real time: (a) Flow chart and (b) description of the algorithm.

in Fig. 12. In this perspective, our methodology can be seen as a twofold control tool for both the optimization of the pricing assignment and the estimation of the users' sensitivity.

At Step 3), the Δ parameter is introduced to allow small perturbations in real time for the measured frequencies of the connection requests $\lambda^m_{h\kappa}(\cdot)$ (with respect to the ideal value $\lambda^o_{h\kappa}$), without updating the NPAF's structure. Within the interval defined by the Δ parameter, the NPAF* maintains suboptimal performance in virtue of the regularity of the optimal solution with respect to small oscillations in the users' sensitivity and in virtue of the well-known generalization capabilities of neural networks.

At Step 4), a new estimation of the demand functions $\lambda_{h\kappa}$ not satisfying the optimal operating condition (A.1) is applied in real time. Since the $\lambda(\cdot)$ are fairly mathematically tractable functions [9], this task is computationally light and can be performed in real time. For instance, in the second simulation scenario, the estimation time horizon T_e was fixed to 60 minutes and the number n^m of samples acquired on the functions $\lambda^m(\cdot)$, driven by video services, was 6. This revealed to be sufficient to approximate the new form of the function $\lambda(\cdot)$, related to video services, with a sufficient degree of accuracy.

At Steps 5) and 6), the new demand functions are computed and the set Ω is updated accordingly. Finally, a new training phase is performed over the updated set Ω. This procedure is faster than the first one, since the just trained NPAF* is employed.

REFERENCES

[1] C. Courcoubetis and R. Weber, *Pricing Communication Networks - Economics, Technology and Modelling*, John Wiley & Sons, San Francisco, CA, 2003.

[2] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," *J. Operat. Res. Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[3] S.H. Low and D.E. Lapsley, "Optimization flow control, I: Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[4] I. Paschalidis and Y. Liu, "Pricing in multiservice loss networks: Static pricing, asymptotic optimality, and demand substitution effects," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 425–437, June 2002.

[5] Q. Wang and J. M. Peha, "State-dependent pricing and its economic implications," *Telecommun. Syst.*, vol. 18, pp. 315–329, 2001.

[6] U. Savagaonkar, E. K. P. Chong, and R. L. Givan, "Online pricing for bandwidth provisioning in multi-class networks," *Computer Netw.*, vol. 44, no. 6, pp. 835–853, Apr. 2004.

[7] S. Kalyanasundaram, E. K. P. Chong, and N. B. Shroff, "Optimal resource allocation in multi-class networks with user-specified utility functions," *Computer Netw.*, vol. 38, no. 5, pp. 613–630, Apr. 2002.

[8] N. J. Keon and G. Anandalingam, "Optimal pricing for multiple services in telecommunications networks offering quality-of-service guarantees," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 60–80, Feb. 2003.

[9] M. Aldebert, M. Ivaldi, and C. Roucolle, "Telecommunications demand and pricing structure: An econometric analysis," *Telecommun. Syst.*, vol. 25, pp. 89–115, 2004.

[10] L. A. Cox, "Data mining and causal modeling of customer behaviors," *Telecommun. Syst.*, vol. 21, pp. 349–381, 2002.

[11] R. Edell and P. Varaiya, "Demand for quality-differentiated network services," in *Proc. CDC*, San Diego, USA, Dec. 1997.

[12] M. Baglietto, R. Bolla, F. Davoli, M. Marchese, and M. Mongelli "A proposal of new price-based call admission control rules for guaranteed performance services multiplexed with best effort traffic," *Comp. Commun.*, vol. 26, no. 13, pp. 1470–1483, 2003.

[13] L. Badia, M. Lindstrom, J. Zander, and M. Zorzi "Demand and pricing effects on the radio resource allocation multimedia communication systems," in *Proc. IEEE Globecom*, San Francisco, CA, Dec. 2003, pp. 139–143.

[14] I. C. Paschalidis and J. N. Tsitsiklis, "Congestion-dependent pricing of network services," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 171–184, Apr. 2000.

[15] X. Lin and N. B. Shroff, "Pricing-based control of large networks," in *Proc. IWDC*, Taormina, Italy, Sept. 2001, pp. 212–231.

[16] W.-S. Kim, "Price-based quality-of-service control framework for two-class network services," *J. Commun. and Netw.*, vol. 9, no. 3, Sept. 2007, pp. 319–329.

[17] P. Xu, M. Devetsikiotis, and G. Michailidis, "Profit-oriented resource allocation using online scheduling in flexible heterogeneous networks," *Telecommun. Syst.*, vol. 31, no. 2–3, pp 289–303, 2006.

[18] K. Ross, *Multiservice Loss models for Broadband Telecommunication Networks*. Springer Verlag, Berlin, 1995.

[19] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Contr. Syst. Mag.*, vol. 22, no. 1, pp. 28–43, Feb. 2002.

[20] E. Altman, D. Artiges, and K. Traore, "On the integration of bestEffort and Guaranteed Performance Services," *Europ. Trans. Telecommun.*, vol. 2, no. 2, Feb.–Mar. 1999.

[21] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, "Call admission control and routing in integrated services networks using neuro-dynamic programming," *IEEE J. Select. Areas Commun.*, vol. 18, no. 2, pp. 197–208, Feb. 2000.

[22] J. H. Lepler and K. Neuhoff, "Resource reservation with a market-based protocol: What prices to expect?," *Comp. Commun.*, vol. 26, pp. 1434–1444, 2003.

[23] InvisibleHand Networks, Inc. [Online]. Available: http://www.invisiblehand.net

[24] N. Semret and A. Lazar, "System and method for performing a progressive second price auction technique," US Patent no. 7,177,832, Feb. 2007.

[25] O. Regev and N. Nisan, "The POPCORN market—an online market for computational resources," in *Proc ACM ICE*, Charleston, SC, USA, Oct. 1998.

[26] Oltsik J. Web services meet the network. IBM White paper. [Online]. Available: https://www14.software.ibm.com

[27] Cisco AON: A network embedded intelligent message routing system. Cisco Systems. [Online]. Available: http://www.cisco.com

[28] M. Poikselka, G. Mayer, H. Khartabil, and A. Niemi, *The IMS IP Multimedia Concepts and Services, 2nd ed.* John Wiley & Sons, LTD, Hoboken, June 2007.

[29] E. W. Fulp and D. S. Reeves, "Bandwidth provisioning and pricing for networks with multiple classes of service," *Computer Netw.*, vol. 46, no. 1, pp. 41–52, 2004.

[30] F. Davoli, M. Marchese, and M. Mongelli, "Neural decision making for decentralized pricing-based call admission control," in *Proc. IEEE ICC*, Seoul, Korea, May 2005, pp. 16–20.

[31] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and extended Ritz method for solution of functional optimization problems," *J. Optim. Theory and Applic.*, vol. 112, no. 2, pp. 403–439, Feb. 2002.

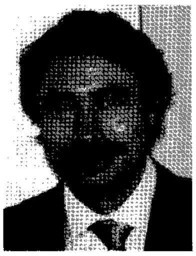
[32] M. Baglietto, T. Parisini, and R. Zoppoli, "Distributed-information neural control: The case of dynamic routing in traffic networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 3, pp. 485–502, May 2001.

[33] S. Chong, S. Li, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[34] [Online]. Available: http://www.bgp-qos.org/forum/

[35] S. Lanning, D. Mitra, Q. Wang, and M. Wright, "Optimal planning for optical transport networks," *Phil. Trans. Royal Soc. London A*, vol. 358, no. 1773, pp. 2183–2196, Aug. 2000.

[36] (1998, Feb.). CFSQP Version 2.5d—Released. [Online]. Available: http://www.isr.umd.edu/Labs/CACSE/FSQP



Franco Davoli (M'90-SM'99) received the Laurea degree in Electronic Engineering in 1975 from the University of Genoa, Italy. Since 1990 he has been Full Professor of Telecommunication Networks at the University of Genoa, where he is with the Department of Communications, Computer, and Systems Science (DIST). From 1989 to 1991 and from 1993 to 1996, he was also with the University of Parma, Italy. His past research activities have included adaptive and decentralized control, large scale systems, routing, and multiple access in packet-switched communication networks and packet radio networks. His current research interests are in bandwidth allocation, admission control, and routing in multiservice networks, wireless mobile and satellite networks, and multimedia communications and services. He has co-authored over 250 scientific publications in international journals, book chapters, and conference proceedings. He is a Member of the Editorial Board of the International Journal of Communication Systems (Wiley), of the international journal Studies in Informatics and Control, and of Simulation–Transactions of the SCS. He has been a Guest Co-Editor of two Special Issues of the European Transactions on Telecommunications and of a Special Issue of the International Journal of Communication Systems. In 2004, he has been the recipient of an Erskine Fellowship from the University of Canterbury, Christchurch, New Zealand, as Visiting Professor. He has been a Principal Investigator in a large number of projects and has served in several positions in the Italian National Consortium for Telecommunications (CNIT). He was the Head of the CNIT National Laboratory for Multimedia Communications in Naples in the years 2003–2004, and Vice-President of the CNIT Management Board for the term 2004–2007. He is a Senior Member of the IEEE.

Mario Marchese (S'94-M'97-SM'04) was born in Genoa, Italy in 1967. He got his Laurea degree cum laude at the University of Genoa, Italy in 1992 and the Qualification as Professional Engineer in April 1992. He obtained his Ph.D. (Italian "Dottorato di Ricerca") degree in "Telecommunications" at the University of Genoa in 1996. From 1999 to 2004, he worked with the Italian Consortium for Telecommunications (CNIT), by the University of Genoa Research Unit, where he was Head of Research. Since February 2005 he has been Associate Professor at the University of Genoa, Department of Communication, Computer and Systems Science (DIST). He is the founder and still the technical responsible of CNIT/DIST Satellite Communications and Networking Laboratory (SCNL) at the University of Genoa. He chaired the IEEE Satellite and Space Communications Technical Committee from 2006 to 2008. He is author and co-author of about 200 scientific works, including international journals, international conferences and book chapters and of the book Quality of Service over Heterogeneous Networks, John Wiley & Sons, Chichester, 2007. His main research activity concerns: satellite and radio networks, transport layer over satellite and wireless networks, quality of service and data transport over heterogeneous networks, and emulation and simulation of telecommunication networks and satellite components.

Maurizio Mongelli got his Laurea degree cum laude at the University of Genoa, Italy in 2000 and the Qualification as Professional Engineer in April 2002. He obtained his Ph.D. (Italian "Dottorato di Ricerca") degree in "Electronic and Computer Engineering" at the University of Genoa in 2004. His Ph.D. was funded by Selenia S.p.A. He worked for both Selenia S.p.A. and the Italian Consortium for Telecommunications (CNIT), at the University of Genoa Research Unit, from 2000 to 2004. He is now a Member of the research staff of the Telecommunication Networking Research Group of the University of Genoa, with a post-doctoral scholarship funded by Selex Communications S.p.A. His main research activity concerns: QoS architectures, resource allocation, and optimization algorithms for telecommunication systems.