Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 2833–2853
Computer Networks
www.elsevier.com/locate/comnet

A new approach to dynamic bandwidth allocation in Quality of Service networks: Performance and bounds ☆

J. Elias [a], F. Martignon [b,*], A. Capone [c], G. Pujolle [a]

[a] The University of Paris 6, LIP6 Laboratory, 8 rue du Capitaine Scott, 75015 Paris, France
[b] The Department of Management and Information Technology, University of Bergamo, Dalmine (BG) 24044, Italy
[c] Department of Electronics and Information of Politecnico di Milano, Italy

Received 8 August 2006; received in revised form 5 December 2006; accepted 6 December 2006
Available online 19 December 2006

Responsible Editor: A. Pitsillides

Abstract

Efficient dynamic resource provisioning algorithms are necessary to the development and automation of Quality of Service (QoS) networks. The main goal of these algorithms is to offer services that satisfy the QoS requirements of individual users while guaranteeing at the same time an efficient utilization of network resources.

In this paper we introduce a new service model that provides per-flow bandwidth guarantees, where users subscribe for a guaranteed rate; moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who can exploit it to transmit at a rate higher than their subscribed rate.

To implement this service model we propose a dynamic provisioning architecture for intra-domain Quality of Service networks. We develop a set of dynamic on-line bandwidth allocation algorithms that take explicitly into account traffic statistics and users' utility functions to increase users' benefit and network revenue.

Further, we propose a mathematical formulation of the extra-bandwidth allocation problem that maximizes network revenue. The solution of this model allows to obtain an upper bound on the performance achievable by any on-line bandwidth allocation algorithm.

We demonstrate through simulation in realistic network scenarios that the proposed dynamic allocation algorithms are superior to static provisioning in providing resource allocation both in terms of total accepted load and network revenue, and they approach, in several network scenarios, the ideal performance provided by the mathematical model.

Keywords: Dynamic bandwidth allocation; Mathematical models; Service model; Utility function

1. Introduction

Efficient dynamic resource provisioning mechanisms are necessary to the development and automation of Quality of Service networks.

☆ Preliminary results of this work have been presented in [1].

* Corresponding author. Tel.: +39 35 205 2357.
E-mail addresses: jocelyne.elias@lip6.fr (J. Elias), fabio.martignon@unibg.it (F. Martignon), capone@elet.polimi.it (A. Capone), guy.pujolle@lip6.fr (G. Pujolle).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2006.12.003

In communication networks, resource allocation is performed mainly in a static way, on time scales on the order of hours to months. However, statically provisioned network resources can become insufficient or considerably under-utilized if traffic statistics change significantly [2]. Telecommunication companies often overbook network resources to increase their income, but overbooking is often performed statically, without considering users' behavior and perceived utility [3].

Therefore, a key challenge for the deployment of Quality of Service networks is the development of solutions that can dynamically track traffic statistics and allocate network resources efficiently, satisfying the QoS requirements of users while aiming at maximizing, at the same time, resource utilization and network revenue.

Recently, dynamic bandwidth allocation has attracted research interest and many algorithms and architectures have been proposed in the literature [2–20].

Several algorithms extend the max–min fair allocation principle [4] taking into account the utility perceived by network applications [5–12]. To implement such allocation algorithms that guarantee QoS constraints to network users, many dynamic provisioning architectures have been proposed [2,13–18]. These approaches and related work are discussed in Section 2.

In this paper, we first propose a new service model that provides quantitative per-flow bandwidth guarantees, where users subscribe for a guaranteed transmission rate. Moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who are willing to pay for it to transmit at a rate higher than their subscribed rate. Bandwidth allocation is enforced at network edges using traffic shaping schemes.

To implement this service model we propose a distributed provisioning architecture composed by core and edge routers. Core routers monitor bandwidth availability and periodically report this information to ingress routers. Moreover, if persistent congestion is detected, core routers notify immediately ingress routers.

Ingress routers perform a dynamic tracking of the effective number of active connections, as proposed in [19,20], as well as of their actual sending rate. Based on such information and that communicated by core routers, ingress routers allocate network resources.

For this purpose, we develop three dynamic on-line bandwidth allocation algorithms, with increasing level of complexity, that take into account traffic statistics as well as users' profile and willingness to acquire extra-bandwidth based on their bandwidth utility function. These algorithms aim at overcoming static provisioning and overbooking techniques by performing resource allocation adaptively and intelligently, maximizing users' satisfaction and improving at the same time network quality.

Further, we propose a mathematical formulation of the extra-bandwidth allocation problem that maximizes network revenue. The solution of this model allows us to obtain an upper bound on the performance achievable by any on-line allocation algorithm.

We evaluate by simulation the performance of our proposed bandwidth allocation algorithms in realistic network scenarios. Numerical results show that our algorithms and service model allow to achieve better performance than statically provisioned networks both in terms of accepted load and network revenue.

In summary, this paper makes the following contributions:

- the definition of a new service model that takes into account users' profile and traffic statistics;
- the proposition of three on-line algorithms for dynamic bandwidth allocation that aim at maximizing both users' utility and network revenue;
- a mathematical formulation of the bandwidth allocation problem that provides an upper bound on the performance achievable by on-line allocation algorithms;
- an extensive performance evaluation of the proposed algorithms in several realistic network scenarios.

The paper is structured as follows: Section 2 discusses related work. Section 3 introduces our proposed service model and provisioning architecture. Section 4 presents a precise statement of the bandwidth allocation problem. Further, a utility-based definition of the network extra-revenue is provided. Section 5 proposes a set of dynamic on-line bandwidth allocation algorithms. Section 6 illustrates a mathematical model for the bandwidth assignment problem that provides upper bounds to our bandwidth allocation algorithms. Section 7 discusses numerical results that show the efficiency of our dynamic bandwidth allocation algorithms compared to a static allocation technique and to the bounds provided by the proposed mathematical model. Finally, Section 8 concludes this work.

2. Dynamic bandwidth allocation: algorithms and architectures

The problem of bandwidth allocation in communication networks has been addressed in many recent works. Both allocation algorithms [4–12] and provisioning architectures [2,13–18] have been proposed in the literature, and are reviewed in the following.

Recently, bandwidth allocation and rate control problems have been addressed in [4–12].

In [4], a max–min fair allocation algorithm is proposed to allocate available bandwidth equally among all connections bottlenecked at the same link. This algorithm implicitly assumes that the utility function, as defined in [21], is the same for each user for a given assigned rate, and therefore it maximizes the minimum bandwidth allocated to each application.

The authors in [5] extend the max–min fair allocation algorithm to the case where each flow can be split among several paths, proposing an approximated algorithm where users' demands are routed and allocated such that the max–min fairness criterion is achieved.

In our work we extend the max–min fair allocation algorithm proposed in [4] to perform a periodic allocation of unused bandwidth, through short-term contracts, to users who are willing to transmit more than their subscribed rate. Note that our approach differs from other ones like the ABR (Available Bit Rate) service in ATM networks [22], or ECN (Explicit Congestion Notification) in IP networks [23], which define asynchronous and continuous rate control mechanisms to avoid network congestion based on router notification messages. Our proposed service model allocates and guarantees extra-bandwidth on demand to users for the duration of at least one update interval, which is on the order of tens of seconds.

The algorithms proposed in [6–12] take explicitly into account application specific utility functions in the bandwidth allocation process.

An extension to the max–min fair allocation algorithm is proposed in [6]. In [7,8], the authors deal with allocation problems in which utility functions are concave.

A distributed approach is proposed in [9], where the authors investigate the problem of allocating transmission data rates to users who have concave as well as sigmoidal utility functions, to take into account the behavior of different applications. Moreover, it is shown that applying rate control algorithms developed for concave utility functions in a generic network scenario can lead to instability and high network congestion.

In [10], the authors extend the Nash Bargaining Solution concept to allocate the bandwidth between applications with general concave utility functions. Furthermore, some computational methods for obtaining fair allocations in a general topology, based on a dual Lagrangian approach and on semi-definite programming, are presented.

In our service model, we do not restrict the utility functions to be concave; in line with [9,10], we allow users to have more general utility functions that are more suitable for modelling the behavior of various applications. However, to solve the problem of extra-bandwidth allocation, we distinguish two cases: (1) all users have concave utility functions [24]; or (2) users have concave as well as non concave utility functions. In the first case, we propose a mathematical formulation of the bandwidth allocation problem that maximizes the total system utility, while in the second case we introduce two on-line measurement-based heuristics.

Bandwidth allocation algorithms are often implemented in network architectures to guarantee QoS constraints to network users. Dynamic bandwidth provisioning in Quality of Service networks has recently attracted a lot of research attention due to its potential to achieve efficient resource utilization while providing the required quality of service to network users [2,13–18].

In [2,13], the authors propose a dynamic edge and core provisioning architecture for differentiated services IP networks. The basic role of edge provisioning is to perform dynamic ingress link sharing. The core provisioning architecture consists of a set of algorithms for interior nodes. A self-adaptive mechanism is adopted to adjust service weights of weighted fair queueing schedulers at core routers; an immediate reduction of edge bandwidth is performed after receiving a Congestion-Alarm signal from core nodes; finally, a periodic bandwidth re-alignment is performed to establish a max–min fair bandwidth allocation to traffic aggregates.

The work in [2] has similar objectives to our research. However, the considered service model differs from our proposed model in that it provides assurances in terms of delay and packet loss bounds using class-based weighted fair queueing schedulers at core routers and various queue management schemes, while in our work we focus mainly on quantitative bandwidth guarantees. Moreover, the traffic statistics are not taken into account in the bandwidth allocation procedure in [2], and only a centralized scheme is considered while in our work we suggest a distributed architecture implementation.

The policy-based architecture presented in [14] for DiffServ networks adjusts dynamically the policies on in-profile and out-of-profile traffics based on the current state of the network estimated using bandwidth monitors. However, this scheme, while achieving dynamic QoS adaptation for multimedia applications, does not take into account the users utility function and their eventual willingness to be charged for transmitting out-of-profile traffic, thus increasing network revenue.

In [15] a connection management strategy for QoS networks is introduced to maximize service providers revenue, while reducing blocking experienced by users.

In [16], a generic pricing structure is presented to characterize the pricing schemes currently used in the Internet, and a dynamic, congestion-sensitive pricing algorithm is introduced to provide an incentive for multimedia applications to adapt their sending rates according to network conditions. As in [16], we take into account users bandwidth utility functions to evaluate our proposed allocation algorithms based on the increased network revenue that is achieved. However, the authors in [16] consider a different service model than that proposed in our work and focus mainly on the issue of dynamic pricing to perform rate adaptation based on network conditions.

The idea of measuring dynamically the effective number of active connections as well as their actual sending rate is a well accepted technique [17,19,20]. In [17], the authors propose an active resource management approach (ARM) for a differentiated services environment. The basic concept behind ARM is that by effectively knowing when a client is sending packets and how much of its allocated bandwidth is being used at any given time, the unused bandwidth can be re-allocated without loss of service. This concept is in line with our research objectives. Differently from our work, however, ARM does not guarantee to the user a minimum subscribed bandwidth throughout the contract duration since unused bandwidth is sent to a pool of available bandwidth and it can be used to admit new connections in the network, in spite of those already admitted.

3. Service model and dynamic provisioning architecture

In this section we first introduce our proposed service model. We then present a distributed provisioning architecture which implements such service model and the signalling messages used to assure the interaction between network elements.

3.1. Service model

We propose a service model that both provides a quantitative per-flow bandwidth guarantee and exploits the unused bandwidth individuated periodically in the network to propose short-term guaranteed extra-bandwidth to users who are willing to pay more to get a higher bandwidth. Our service model allocates network resources adaptively and intelligently, allowing network quality differentiation and improving resource utilization even when bandwidth overbooking is deployed.

In the allocation process, different weights can be assigned to network users to distribute extra-bandwidth with different priorities; such weights can be set statically off-line, based on the service contract proposed to the user, or can be adapted on-line based, for example, on the user bandwidth utility function.

Our proposed service model is therefore characterized by:

- A quantitative bandwidth guarantee, expressed through the specification of user's subscribed rate.
- Short-term guaranteed extra-bandwidth: the network is monitored on-line to individuate unused bandwidth that is allocated to users who can exploit it to transmit extra-traffic for the duration specified in the contract. The contract duration depends on the user request and can span over several consecutive update intervals.
- A weight that is used in the extra-bandwidth assignment procedure and is related to the global benefit of assigning additional bandwidth to that user.

Note that part of the unused bandwidth can be collected from users that transmit at a rate inferior to their subscribed rate. However, when such users increase their transmission rate, they will obtain at least their subscribed rate within few iterations of the bandwidth allocation algorithm. In these situations, the provider could pay a penalty for the period in which the user has been assigned a rate less than his subscribed rate.

Throughout this paper, we will consider the case with the minimum possible contract duration, i.e. one update interval. This will provide a basis for comparing the performance of all the proposed bandwidth allocation algorithms, since it provides the maximum flexibility in bandwidth allocation.

Bandwidth allocation is enforced at network edges using standard traffic shaping techniques implemented at ingress nodes. Inside the network we assume that flow classification mechanisms and QoS-based scheduling schemes are adopted for each micro-flow or at least for flow aggregates. MPLS [25] and GMPLS [26,27] networks with advanced scheduling techniques like WFQ [28] are one of the most suitable application scenarios for the proposed service model.

Finally, note that our proposed service model and allocation algorithms can be applied to micro-flows or traffic aggregates, depending on the application scenarios and the granularity of traffic control enforced in the network. An interesting network service that can benefit from the proposed model is Virtual Private Network, where traffic is quite variable and both QoS perceived by users and network utilization can be greatly improved managing dynamically network resources.

3.2. Architecture and control messaging

To implement our service model we assume a distributed architecture constituted by core and edge routers, as shown in Fig. 1; traffic monitors are installed on ingress and core routers to perform on-line measurements on the incoming traffic flows and links utilization, respectively.

Each ingress router can measure the offered rate as well as the transmission rate of each connection entering the network through it, defined respectively as the total load generated by the connection and the amount of its traffic accepted in the network. Such information is then exchanged periodically, through update messages, with all the other ingress routers to report the current incoming traffic statistics.

Every update interval, all ingress routers execute simultaneously the same algorithm to individuate unused and spare bandwidth, and to allocate such resources dynamically and efficiently based on local information as well as on traffic statistics contained in the received update messages. For this purpose we propose in Section 5 three bandwidth allocation algorithms with increasing complexity and performance. Bandwidth allocation is then

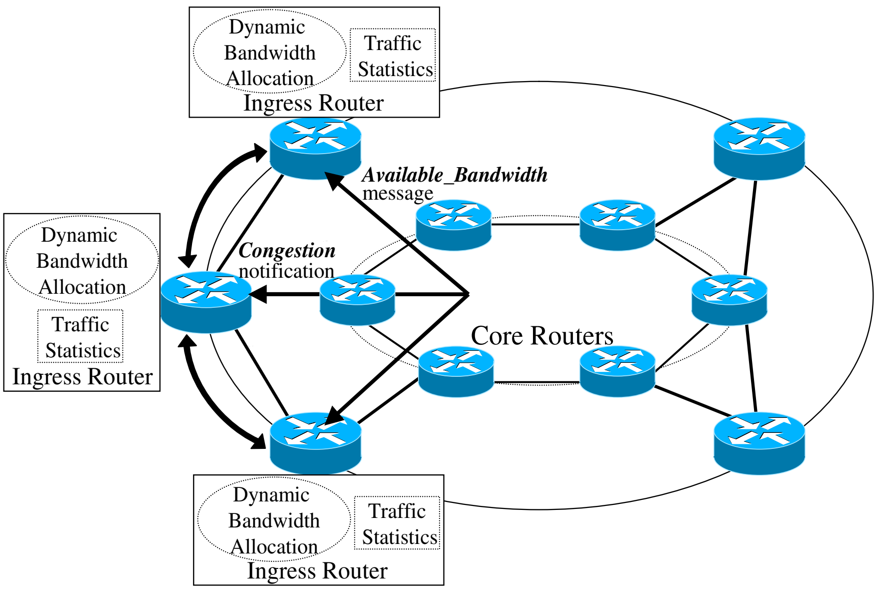

Fig. 1. The proposed distributed architecture that supports dynamic bandwidth allocation.

enforced using traffic shaping techniques at ingress routers.

Note that, since bandwidth allocation is performed only every update interval, traffic variations can be tracked only with such granularity. Obviously, a trade-off exists between the granularity in bandwidth allocation and the overhead resulting from more frequent message exchanges. We will provide an upper bound on the overhead due to the exchange of signalling messages as well as on the delay involved in updating such information in Section 7, showing that our proposed approach is viable in realistic network scenarios.

Since each ingress router executes the same allocation algorithm autonomously, the allocation procedure always terminates. To achieve the best performance all ingress routers should ideally work synchronously. If some ingress routers execute the allocation algorithm with out-of-date network state information it may happen that more traffic is admitted than the network can afford, causing congestion at core routers.

Therefore, in our architecture we assume that core routers measure links utilization and provide a feedback to ingress routers using techniques similar to ECN [23]. Such feedback is used to notify the ingress routers that congestion is experienced in some network nodes, or that link and node failures are detected, so that ingress routers can invoke immediately bandwidth re-allocation to solve these situations.

The messages exchanged between network routers, illustrated with arrows in Fig. 1, are similar to the control messages proposed in [2] to report persistent congestion or resource availability. A subset of the messages defined in the RNAP protocol [29] can be used for these purposes.

We observe that, since in our architecture we consider update intervals on the order of tens of seconds, precise synchronization between edge routers is not a key issue. We discuss in more details the trade-offs in the setting of the update interval in Section 7.

Since dynamic provisioning algorithms are complementary to admission control algorithms [2], in our work we assume that admission control algorithms are adopted at the edge of the network. Admission control algorithms guarantee that the problem of assigning the minimum required bandwidth is always feasible and that all the spare bandwidth can be exploited.

Finally, note that a centralized architecture that implements our proposed service model can be devised as well; the extension with respect to the proposed distributed architecture is straightforward and therefore is not discussed in this paper.

4. Network model

Let us model the network as a directed graph $G = (N, L)$ where nodes represent routers and directed arcs represent links. Each link $l \in L$ has associated the capacity C_l. A set of K connections is offered to the network. Each connection is represented by the notation $(s_k, d_k, sr_k, r\_min_k)$, for $k = 1, \ldots, K$, where s_k, d_k, sr_k and r_min_k represent respectively the connections source node, destination node, subscribed rate and the minimum bandwidth the application requires to work correctly. Let a_k^l be the routing matrix: $a_k^l = 1$ if connection k is routed on link l, $a_k^l = 0$ otherwise. We assume that a communication between a user pair is established by creating a session involving a path that remains fixed throughout the user pair conversation duration. The session path choice method (i.e. the routing algorithm) is not considered in this paper.

We denote the interval between two successive allocations performed by the algorithm as the *update interval*, whose duration is T_u seconds.

At the beginning of every update interval n, each ingress router determines, for every connection k that enters the network through it, the offered load (Ol_k^{n-1}) as well as its actual transmission rate (b_k^{n-1}), obtained respectively by averaging over the previous update interval the total load generated by connection k and the amount of its traffic accepted in the network.

This information is then sent to all other ingress routers using control messages as described in the previous section, so that all ingress routers share the same information about current traffic statistics and perform simultaneously the same allocation procedure outputting the same assigned rates for the connections.

Based on this information, connections are classified according to their traffic statistics, as specified in the following. Such classification is used in the bandwidth allocation process to achieve high network revenue.

4.1. Connections classification

The K connections offered to the network are first classified according to the following criteria.

Connections having $b_k^{n-1} < r\_min_k$ are considered *idle*; all other active connections are further classified as *greedy* if they used all their subscribed rate sr_k, otherwise they are classified as *non-greedy*.

We denote by K_i, K_{ng} and K_g the sets of *idle*, *non-greedy* and *greedy* connections, respectively.

Greedy connections are assigned extra-bandwidth by the network according to their traffic statistics and utility functions, thus contributing to increase network revenue, defined in the following.

Note that some owners of greedy connections may not be inclined to pay more to transmit at a rate higher than their subscribed rate. However, it is easy to take into account users' indications since each user can specify in his contract (either statically or dynamically) whether he is willing to participate in the extra-bandwidth allocation process or not.

Note that connections are classified based on their transmission rate only. In fact, in this classification, we do not consider the connections' offered rate since it may be difficult to measure it for all sources; this is the reason why one of our proposed bandwidth allocation algorithms does not take into account this parameter.

4.2. Network revenue

We define, in line with [16], the average network revenue as the total charge paid to the network for all the extra-bandwidth utilization, averaged over all the bandwidth update intervals. In this computation we consider only network revenue generated by greedy users that are assigned extra-bandwidth by our proposed dynamic allocation algorithms. The revenue deriving from the static subscription of guaranteed rates (sr_k) is not considered since we focus on the extra network revenue that dynamic allocation can generate.

As already stated, we do not consider the pricing component of bandwidth utility functions, and we simply assume that network revenue is equal to the extra-utility perceived by network users. Furthermore we assume, in line with [7], that the utilities are additive so that the aggregate utility of rate allocation is given by the sum of the utilities perceived by all network users.

Using the notation introduced in the previous section, the average network extra-utility can be obtained averaging over all the update intervals i the quantity:

$$\sum_{k\in K_g^i} U_k(b_k^i) - U_k(sr_k), \tag{1}$$

where sr_k and b_k^i represent, respectively, the subscribed rate and the average transmission rate in the ith update interval for all the connections $k \in K_g^i$ that are greedy in such interval, and $U_k(x)$ models the perceived utility of the kth user for an allocation of x bandwidth units.

Since in this paper we assume for simplicity that the price paid by each greedy user is equal to its perceived extra-utility, (1) represents also the network extra-revenue. In all the numerical results presented in Section 7 we use the expression ''network extra-revenue'' with such meaning.

Note that the goal of our proposed bandwidth allocation algorithms is different from that of previous works [4–12] since we focus on maximizing the network extra-revenue. The algorithms presented in the next Section consider the connections classification as a basic step of the bandwidth allocation process.

The maximization of the extra-revenue (1) can be further interpreted from a different point of view: if we consider network scenarios where users are charged based only on their subscribed rate (e.g. flat charging in xDSL access networks), our allocation algorithms increase the overall users' utility, thus improving the quality of network services perceived by all users. Therefore, networks where the proposed algorithms are deployed will provide a better quality of service to their users.

5. Dynamic bandwidth allocation algorithms

In this section, we propose three dynamic bandwidth allocation algorithms with different complexity and performance.

The proposed algorithms take into account the traffic statistics measured on-line, the links utilization and users' utility functions to allocate network capacity maximizing the total network revenue and satisfying at the same time the individual user's requirements. The goal is to determine r_k^n, the amount of bandwidth allocated to each source k in the nth update interval.

All the proposed algorithms proceed in two main steps:

- In step one, bandwidth is allocated to all active connections trying to match their near-term traffic requirements that are predicted based on the statistics collected by ingress routers.
- In step two, the spare bandwidth as well as the bandwidth left unused by idle and active

connections is individuated on each link. Such available extra-bandwidth is allocated with guarantee during the current update interval exclusively to connections that can take advantage of it since they are willing to transmit at a rate higher than their subscribed rate.

The first step (Section 5.1) is in common with all the allocation algorithms, while the second step (Section 5.2) differs for each algorithm as detailed in the following.

5.1. Bandwidth allocation according to user requirements

Using the definitions and notation introduced previously, we perform the following assignments:

- *Idle* connections are assigned their minimum required transmission rate, i.e. $r_k^n = r\_\min_k$, $\forall k \in K_i$.
- *Non-greedy* connections are assigned a bandwidth that can accommodate traffic growth in the current update interval while, at the same time, save unused bandwidth that can be re-allocated to other users.
  Several techniques have been proposed in the literature to predict the near-term transmission rate of a connection based on past traffic measurements; however, the focus of this paper is not to determine the best traffic predictor.
  In this work we only consider the last measured transmission rate, b_k^{n-1}, and we propose the following simple bandwidth allocation: $r_k^n = \min\{2 \cdot b_k^{n-1}, sr_k\}, \forall k \in K_{ng}$. Hence, the rate assigned in this step to each *non-greedy* connection is at most the double of the rate used in the previous interval without exceeding the subscribed rate which is guaranteed to such connection.
  We will gauge the impact of this choice in Section 7, based on the performance achieved by our proposed mathematical model.
  Since the bandwidth allocated to a user can at most double from one update interval to the other, this could affect the performance of users that experience steep increase in their subscribed rate. For this reason, a penalty can be introduced when the user perceives that he is assigned less bandwidth than his subscribed rate.
- *Greedy* connections are assigned in this step their subscribed rate sr_k, and they further take part to the allocation of extra-bandwidth performed in the second step, since they are already exploiting all their subscribed rate.

5.2. Allocation of spare bandwidth

After having performed the allocations described in the first step, each algorithm individuates on each link l the residual bandwidth R_l, i.e. the spare bandwidth as well as the bandwidth left unused by idle and non-greedy connections. R_l is hence given by the following expression:

$$R_l = C_l - \left(\sum_{k \in K_i \cup K_{ng}} r_k^n \cdot a_k^l + \sum_{k \in K_g} sr_k \cdot a_k^l \right), \quad \forall l \in L, \tag{2}$$

where the first summation represents the total bandwidth allocated in the first step to idle and non-greedy connections, while the second summation represents the bandwidth allocated to greedy connections. Such extra-bandwidth is distributed exclusively to greedy users who can exploit it to transmit at a rate higher than their subscribed rate.

To this aim, we propose in the following three dynamic on-line algorithms for the bandwidth allocation problem; in Section 6 we illustrate a mathematical model which provides an upper bound to the network revenue that can be obtained by any dynamic on-line bandwidth allocation scheme.

5.2.1. Optimum bandwidth allocation algorithm

If we assume that the utility function $U_k(x)$ and the offered load are known or can be estimated for every greedy source k, then we can illustrate the *Optimum Bandwidth Allocation* (OBA) problem formulation. The decision variable f_k^n, $k \in K_g$, represents the amount of extra-bandwidth that is allocated to each greedy connection during the nth update interval. We maximize the total network extra-revenue considering the following mathematical model:

$$\text{Maximize} \quad \sum_{k \in K_g} U_k(sr_k + f_k^n) - U_k(sr_k), \tag{3}$$

$$\text{s.t.} \quad \sum_{k \in K_g} f_k^n \cdot a_k^l \leqslant R_l, \quad \forall l \in L, \tag{4}$$

$$sr_k + f_k^n \leqslant Ol_k^{n-1}, \quad \forall k \in K_g, \tag{5}$$

$$f_k^n \geqslant 0, \quad \forall k \in K_g. \tag{6}$$

The objective function (3) is the total network extra-revenue.

Constraint (4) represents capacity constraints expressed for each link of the graph.

Constraint (5) imposes that, for every update interval n, the total load allocated to each greedy source k does not exceed the total load offered to the network by k, thus avoiding to waste extra-bandwidth.

Here we assume that the traffic offered by each connection in the current update interval can be approximated by that measured in the previous interval. With such choice we implemented a simple traffic prediction strategy, and we will demonstrate in Section 7 that OBA can achieve a significant performance improvement with respect to other allocation algorithms even with such simplistic assumption. Evidently, more efficient traffic predictors can be devised to increase the efficiency of OBA, but this is out of the main focus of this paper.

This problem formulation is in general non-linear due to utility functions; it involves a number of decision variables equal to the number of greedy connections $|K_g|$, which can be at most equal to the total number of connections offered to the network, K. On the other hand, the number of constraints is equal to $|L| + 2|K_g|$, where $|L|$ is the total number of links in the network.

If sources' offered load is unknown or difficult to estimate, we can consider an alternate formulation to the OBA problem by simply dropping constraint (5).

If all users' utility functions are differentiable and strictly concave, then the objective function (3) is differentiable and strictly concave. Since the feasible region (4)–(6) is compact, a maximizing value of f_k^n exists and can be found using Lagrangian methods.

When utility functions are unknown or even offered loads cannot be measured, the OBA model can not be applied. For this reason we further develop two other heuristic algorithms, detailed in the following.

5.2.2. Simple dynamic bandwidth allocation algorithm

When neither the utility functions nor the offered loads can be easily measured for all sources, we propose the Simple Dynamic Bandwidth Allocation (SDBA) algorithm, recently presented by the authors in [1] and detailed in Table 1. SDBA is an extended version of the max–min fair allocation algorithm introduced in [4], where the extra-bandwidth is distributed to greedy sources according to the max–min fairness criterion.

Table 1
Pseudo-code specification of the Simple Dynamic Bandwidth Allocation algorithm (SDBA)

	Algorithm SDBA(K_g, R_l, a_k^l)
(1)	initialize all $f_k^n = 0, \forall k \in K_g$
(2)	remove from the link set L all links $l \in L$ that have a number of connections crossing them n_l equal to 0
(3)	for every link $l \in L$, compute $F_l = R_l/n_l$
(4)	identify the link α that minimizes F_α i.e. $\alpha \| F_\alpha = \min_k(F_k)$
(5)	set $f_k^n = F_\alpha, \forall k \in K_\alpha$, where $K_\alpha \subseteq K_g$ is the set of greedy connections that cross link α
(6)	for every link l, update the residual capacity and the number of crossing greedy connections as follows: $R_l = R_l - \sum_{k \in K_\alpha} f_k^n \cdot a_k^l$ $n_l = n_l - \sum_{k \in K_\alpha} a_k^l$
(7)	remove from set L link α and those that have $n_l = 0$
(8)	if L is empty, then stop; else go to Step (3)

The simple idea behind SDBA is the following: the extra-bandwidth is allocated equally to all greedy sources bottlenecked at the same link, starting from the link that imposes the most stringent capacity constraint.

SDBA takes as input the set K_g of greedy connections, the link set L with the residual capacity on each link l, R_l, and the routing matrix a_k^l, and produces as output the amount of extra-bandwidth $f_k^n, k \in K_g$ that is assigned to each greedy connection during the nth update interval, so that finally $r_k^n = sr_k + f_k^n, \forall k \in K_g$.

To illustrate the operation of SDBA, let us refer to the simple network scenario shown in Fig. 2, where four greedy connections are active in the nth update interval: two connections (C_1 and C_2)

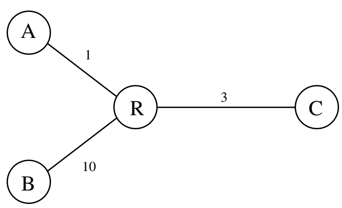

Fig. 2. Example scenario that illustrates the operation of the SDBA algorithm: two connections are established between nodes (A, C) and between nodes (B, C). Residual capacities are indicated next to each link.

Table 2
Path for the connections in the example scenario of Fig. 2

Connection	Path
C_1	$A-R-C$
C_2	$A-R-C$
C_3	$B-R-C$
C_4	$B-R-C$

between nodes (A, C) and two connections (C_3 and C_4) between nodes (B, C). Connections paths are reported in Table 2, while the residual capacity R_l, expressed in bandwidth units, is indicated for each link in Fig. 2.

In the first iteration, F_l is equal to 0.5 for link $A-R$, to 5 for link $B-R$ and to 0.75 for link $R-C$; hence the first bottleneck is link $A-R$, $F_\alpha = 0.5$ and $f^n_{C_1} = f^n_{C_2} = 0.5$. The residual capacities on links $A-R$ and $R-C$ become equal to 0 and 2, respectively. In the second iteration, F_l is equal to 1 for link $R-C$ and to 5 for link $B-R$; hence the second bottleneck is link $R-C$, $F_\alpha = 1$ and $f^n_{C_3} = f^n_{C_4} = 1$.

5.2.3. Iterative dynamic bandwidth allocation algorithm

Since SDBA does not take into account sources' offered load, it can allocate to a source more bandwidth than it can use, thus wasting it. For this reason, we propose the Iterative Dynamic Bandwidth Allocation (IDBA) algorithm, detailed in Table 3, that takes into account source's offered load, which can be measured on-line at ingress routers. IDBA can therefore be used when users' utility functions are unknown while sources' offered load can be determined.

IDBA takes as input the offered load Ol_k^{n-1}, $k \in K_g$ besides all the inputs considered by SDBA, to compute the amount of extra-bandwidth $f_k^n, k \in K_g$ that is assigned to each greedy connection during the nth update interval.

In this algorithm, a greedy connection is defined as *Served* when its total assigned rate, $sr_k + f_k^n$, is at least equal to its offered load, otherwise it is defined as *non-Served*. Let K_g^S and K_g^{NS} denote the set of greedy *Served* and *non-Served* connections respectively.

For every update interval n, all greedy connections $k \in K_g$ are initialized as *non-Served*. In every iteration, the SDBA algorithm is used to determine the extra-flow assignment for all *non-Served* connections.

Table 3
Pseudo-code specification of the Iterative Dynamic Bandwidth Allocation algorithm (IDBA)

```
      Algorithm IDBA(K_g, Ol_k^{n-1}, R_l, a_k^l)
(1)   initialize K_g^{NS} = K_g
(2)   run SDBA(K_g^{NS}, R_l, a_k^l), thus determining f_k^n, k ∈ K_g^{NS}
(3)   set N_{g,b}^{NS} = 0
      where N_{g,b}^{NS} represents the number of bottlenecked
      connections
      among the set of greedy non-Served connections, K_g^{NS}
(4)   for every connection k ∈ K_g^{NS},
        if (sr_k + f_k^n < Ol_k^{n-1})
          increase N_{g,b}^{NS} by 1
        end if
      end for
(5)   if (N_{g,b}^{NS} = |K_g^{NS}|)
          Mark all connections as served; then stop
        end if
(6)   if (N_{g,b}^{NS} < |K_g^{NS}|)
        for every connection k ∈ K_g^{NS},
          if (sr_k + f_k^n ≥ Ol_k^{n-1})
            r_k^n = Ol_k^{n-1}
            f_k^n = r_k^n − sr_k
            mark k as served; hence K_g^{NS} = K_g^{NS} − {k}
            for every link l, update the residual capacity and
            the number of crossing greedy connections as
            follows:
                R_l = R_l − Σ_k f_k^n · a_k^l
                n_l = n_l − Σ_k a_k^l
            end for
          end if
        end for
      end if
(7)   if (All connections are served) or (There is no extra-
      bandwidth to distribute)
      then stop;
      else go to Step (2)
      end if
```

IDBA iterates over the two following steps until either all sources become *Served* or the extra-bandwidth on the bottleneck links is completely allocated:

- First, the extra-bandwidth individuated in the network is allocated fairly among the *non-Served* greedy connections using SDBA(K_g^{NS}, R_l, a_k^l).
- Second, for every *non-Served* greedy connection, $k \in K_g^{NS}$ if $sr_k + f_k^n > Ol_k^{n-1}$ then r_k^n is set equal to Ol_k^{n-1} and the connection is classified as *Served*; the difference, $sr_k + f_k^n - Ol_k^{n-1}$, is redistributed in the next iteration to the remaining *non-Served* greedy connections.

To take into account users weights in the SDBA and IDBA algorithms, it is sufficient to substitute n_l

in Tables 1 and 3 with w_l, which is defined as the sum of the weights of all greedy connections that are routed on link l.

It should be clarified that our proposed algorithms can temporarily present some limitations in bandwidth allocation, since the bandwidth allocated to a user can at most double from an update interval to the successive one. This could affect the performance of users that experience steep increases in their transmission rate. In these situations, as already stated, a penalty can be introduced if the user perceives that his subscribed rate has not been allocated in the current update interval.

In Section 7 we evaluate numerically this effect showing at the same time how it is counterbalanced by increased network revenue in all the considered network scenarios under several traffic load conditions.

6. Ideal bandwidth allocation algorithm

In this section we introduce a novel and flexible mathematical model, Ideal Bandwidth Allocation (IBA), that maximizes the total network revenue and provides bounds to the performance achievable by any on-line dynamic bandwidth allocation algorithm.

IBA assumes the exact knowledge of future traffic offered to the network. We maintain all the assumptions made for the OBA model in the previous section. The bandwidth allocation is performed optimizing the operation of the network loaded with the actual and future traffic. No practical bandwidth allocation scheme can perform better.

This mathematical model extends the utility maximization problem addressed in [7] and its solution can be obtained with LP techniques when all utility functions are concave.

IBA allows to gauge the impact of the traffic predictor used to assign bandwidth to *non-greedy* connections, as well as of the granularity in bandwidth assignment (i.e. the sensitivity to the parameter T_u) on the performance of our proposed dynamic bandwidth allocation algorithms.

Let us model the network with the same notation introduced previously, where a set of K connections is offered to the network. Each connection k is further characterized by its arrival time, its duration and the traffic offered to the network in every instant τ, $Ol_k(\tau)$. Given K connections (Fig. 3 shows an example for $K = 4$), the time interval from the arrival of the first connection and the ending time of the last connection is subdivided in a set I of time intervals. In each time interval, t, the traffic offered by each connection (referred to hereafter as Ol_k^t) remains constant.

In each time interval t, all connections that have $Ol_k^t < sr_k$ (i.e. *idle* and *non-greedy* connections) are assigned their offered load Ol_k^t. All other connections are classified as *greedy*; let K_g^t be the set of connections that are *greedy* in time interval t. Let R_l^t be the residual bandwidth available on each link l in time interval t after having allocated bandwidth to all *idle* and *non-greedy* connections.

Based on the above definitions and notations, we establish the mathematical formulation of the IBA model. The decision variable r_k^t, $k \in K_g^t$, represents the amount of bandwidth that is allocated to each

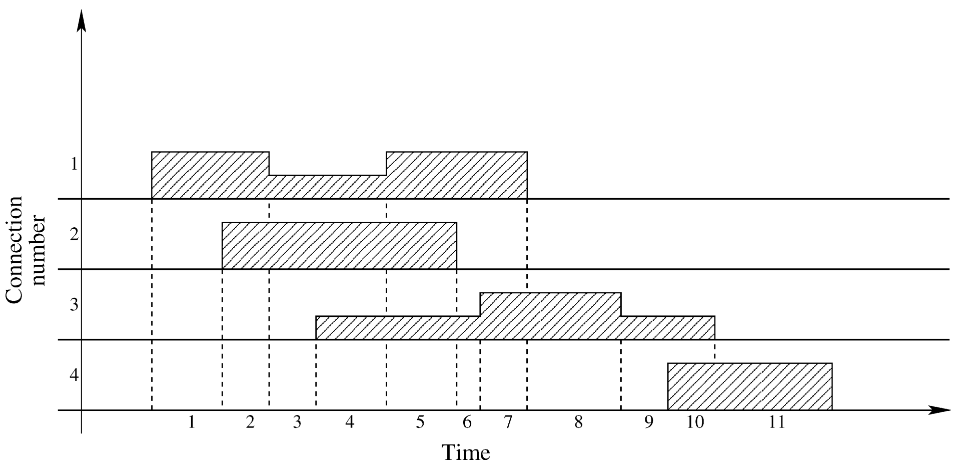

Fig. 3. Arrival time, offered load and duration of the connections offered to the network.

greedy connection during the time interval t. We maximize the total network extra-revenue considering the following mathematical model:

$$\text{Maximize} \quad \sum_{t \in I} \sum_{k \in K_g^t} U_k(r_k^t) - U_k(sr_k), \tag{7}$$

$$\text{s.t.} \quad sr_k \leqslant r_k^t \leqslant Ol_k^t, \forall k \in K_g^t, \quad t \in I, \tag{8}$$

$$\sum_{k \in K_g^t} r_k^t \cdot a_k^l \leqslant R_l^t, \quad \forall l \in L, \ t \in I. \tag{9}$$

The objective function (7) is the total network extra-revenue.

Constraint (8) imposes that, for every time interval t, the total load allocated to each *greedy* source k does not exceed the total load offered to the network by k, thus avoiding to waste extra-bandwidth.

Finally, constraint (9) represents capacity constraints expressed for each link of the graph.

7. Numerical results

In this section we compare the performance of the proposed dynamic bandwidth allocation algorithms with a static provisioning strategy, which consists in assigning to each source k its subscribed rate sr_k, and to the theoretical bounds provided by the proposed mathematical model.

We refer to different network scenarios to cover a wide range of possible environments. The simulation tool we used was J-Sim simulator version 1.3 [30].

We consider the following performance metrics: the average accepted load and network extra-revenue. The average accepted load is obtained averaging the total load accepted in the network over all the bandwidth update intervals, while the average network extra-revenue has been defined in Section 4.

All numerical results have been calculated over long-lived data exchanges, achieving very narrow 95% confidence intervals [31]. The duration of each simulation was equal to 5000 s.

In the first scenario we gauge the effectiveness of the proposed traffic-based bandwidth allocation algorithms. We consider, in line with [2,16], the scenario illustrated in Fig. 4, that consists of a single bottleneck with 2 core nodes, 6 access nodes and 40 end nodes (20 source–destination pairs). All links are full-duplex and have a propagation delay of 1 ms.

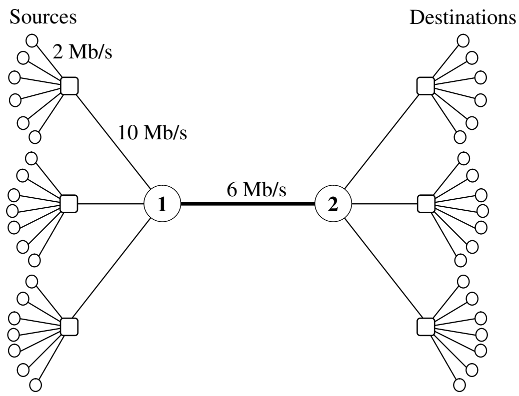

Fig. 4. Network topology with a single bottleneck and 20 source–destination pairs.

The capacity of the link connecting the two core nodes is equal to 6 Mb/s, that of the links connecting the access nodes to core nodes is equal to 10 Mb/s, and that of the links connecting the end nodes to access nodes is 2 Mb/s. The buffer size of each link can contain 50 packets. All these settings are in line with those used in [2] to obtain numerical results.

We use 20 exponential On–Off traffic sources; the average On time is set to 200 s, and the average Off time is varied in the 0–150 s range to simulate different traffic load conditions while varying at the same time the percentage of bandwidth left unused by each connection. During On times each source transmits with a fixed rate that we refer to hereafter as the source's peak rate.

Six sources have a peak rate of 50 kb/s and a subscribed rate of 150 kb/s, eight sources have a peak rate of 250 kb/s and a subscribed rate of 200 kb/s, while the remaining six sources have a peak rate of 1 Mb/s and a subscribed rate of 500 kb/s; the minimum bandwidth required by each source, r_min_k, is equal to 10 kb/s. The algorithm updating interval, T_u, is set to 20 s.

We assume, for simplicity, that all users have the same weight and the same utility function proposed in [8], $U_k(x) = a \cdot \log(b + x)$, where $a \geqslant 0$ and $0 \leqslant b \leqslant 1$. In this scenario, we consider $a = 0.5$ and $b = 1$.

Note that a realistic characterization of network applications is outside the scope of this paper. The specification of the utility function allows us exclusively to gauge the extra network revenue that can derive from the deployment of our proposed bandwidth allocation algorithms.

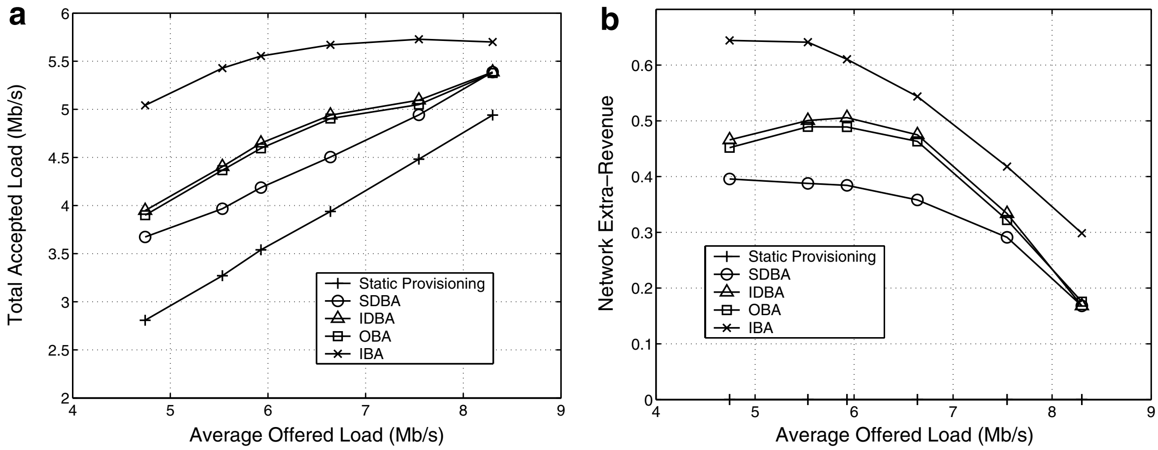

Fig. 5. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Fig. 4 (all users have the same utility function).

Fig. 5a and b shows, respectively, the average total load accepted in the network and the corresponding total extra-revenue as a function of the average total load offered to the network.

In all the following figures we report the upper bound provided by our proposed mathematical model, IBA. We compare the performance of our allocation algorithms to such bound later in this section.

It can be observed that the best performance is achieved by the Optimum Bandwidth Allocation algorithm, OBA, and the Iterative Dynamic Bandwidth Allocation algorithm, IDBA. SDBA achieves lower performance, but it still outperforms static bandwidth allocation both in terms of total accepted load and network revenue.

The maximum network extra-revenue is achieved when the average Off time of exponential sources is equal to 80 s, corresponding to an offered load approximately equal to 6 Mb/s. In this situation, in fact, the average number of idle connections (i.e. 6) is sufficiently high to exalt our dynamic allocation algorithms that re-allocate unused bandwidth to active users who can take advantage of it, sending extra-traffic and generating network extra-revenue. With lower Off time values (i.e. with higher offered loads) the total revenue slightly decreases as less connections are idle, in average, and consequently less bandwidth is available for re-allocation.

To investigate the impact of the update interval duration on the algorithms' performance, we have considered, in the same scenario, different values for T_u, viz. 40 s and 60 s. We found that the average increase in the total accepted load, expressed as a percentage of the traffic admitted in the static allocation case, is of 14% for $T_u = 40$ s and 11% for $T_u = 60$ s, while for $T_u = 20$ s it was 24% (see Fig. 5a).

Using small values for T_u allows to track traffic variations more precisely, thus leading to a better performance. However, there exists a trade-off between the granularity in bandwidth allocation and the overhead resulting from more frequent message exchanges, as we will quantify at the end of this section.

In the same scenario of Fig. 4 we then considered different utility functions for some sources. More specifically, the six connections having a subscribed rate equal to 500 kb/s have associated the utility function $\frac{3}{2} \cdot \log(1 + x)$, while all the other connections have the same utility function as in the previous scenario, i.e. $\frac{1}{2} \cdot \log(1 + x)$.

Fig. 6a and b shows the total accepted load and network extra-revenue achieved by the various allocation algorithms. In this scenario, OBA achieves the better performance in terms of network revenue. This is expected since it distributes network extra-bandwidth taking into account users' utility functions, differently from IDBA, SDBA and static provisioning.

Finally, in the same scenario of Fig. 4 we fixed the average Off time of exponential sources to 100 s while maintaining the average On time equal to 200 s, and we varied the peak rate of all sources scaling it by a factor α, with $0.25 \leqslant \alpha \leqslant 1.5$. We

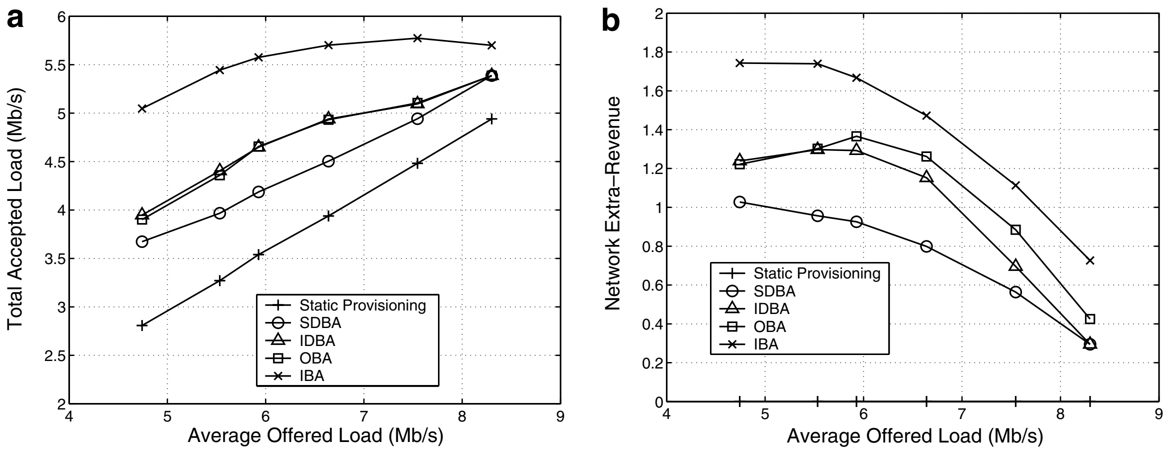

Fig. 6. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Fig. 4. Eight sources have utility function $U_1(x) = \frac{3}{2} \cdot \log(1 + x)$, while all the others have associated $U_2(x) = \frac{1}{2} \cdot \log(1 + x)$.

considered different utility functions as in the previous scenario. Fig. 7a and b shows the total accepted load and the total extra-revenue in this scenario.

At very low load the static provisioning technique achieves slightly higher performance than dynamic allocation algorithms, and it practically overlaps the upper bound provided by our proposed mathematical model, IBA. This is due to the fact that in this situation static provisioning is in effect sufficient to accommodate all incoming traffic; on the other hand, dynamic provisioning algorithms need some time (in the worst case up to T_u seconds) to track the transition of sources from the idle to the active state. For all other traffic loads the advantage of the proposed dynamic bandwidth allocation algorithms with respect to static allocation is evident both in terms of accepted load and network revenue. More specifically, OBA achieves the best performance in terms of network revenue especially for high offered loads; note that IDBA achieves almost the same performance as SDBA for medium offered loads.

We further considered a scenario where bandwidth allocation is applied to aggregate traffic flows. To this aim we used the same network topology of Fig. 4, with 18 end nodes (nine source–destination pairs). We considered nine aggregate traffic sources which generate long-range dependent (LRD) network traffic obtained, in line with [32], by multiplexing 10 Pareto On–Off sources with average On time set to 200 s, and average Off time varied in the 0–300 s range. The Pareto shape parameter of both

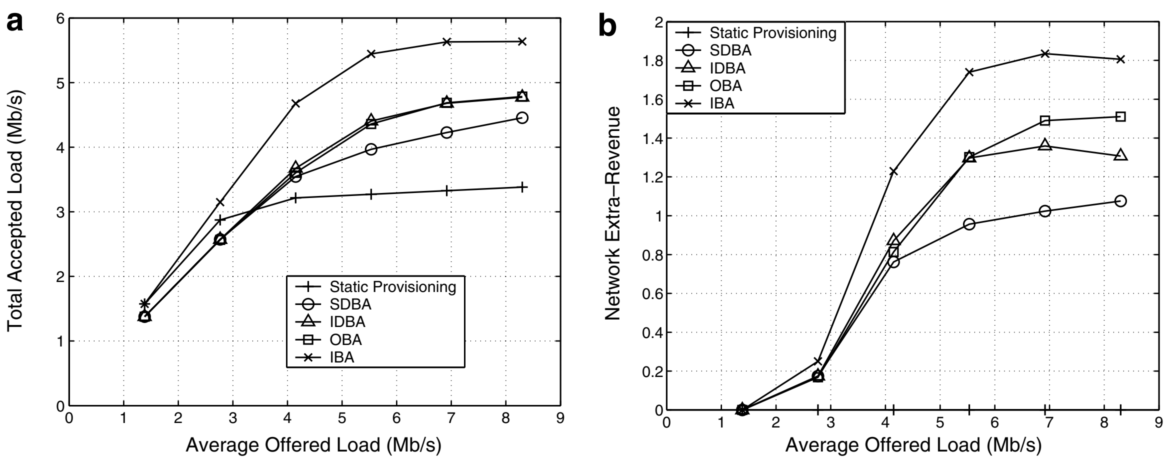

Fig. 7. Average total accepted load (a) and network extra-revenue (b) as a function of the average load offered to the network of Fig. 4. The Off time of the exponential sources is fixed and set to 100 s.

On and Off periods was set to 1.9. Three out of nine aggregate sources have a peak rate of 250 kb/s, three have a peak rate of 1 Mb/s and the remaining three have a peak rate of 2 Mb/s. All aggregate sources have a subscribed rate of 500 kb/s. The sources with a peak rate of 2 Mb/s have associated the utility function $\frac{3}{2} \cdot \log(1 + x)$, while all the other sources have a utility function equal to $\frac{1}{2} \cdot \log(1 + x)$. The results are shown in Fig. 8a and b and confirm that our proposed bandwidth allocation algorithms provide good performance also in the presence of aggregate traffic flows, with OBA exhibiting the best performance. Even though for high offered loads the traffic accepted is almost the same for all the considered allocation algorithms, OBA achieves higher network extra-revenue since it takes explicitly into account users' utility functions in the extra-bandwidth distribution process. These results are in line with those reported in Fig. 6a and b.

We then considered the network topology shown in Fig. 9, originally proposed in [2]. It comprises eight core nodes and seven bidirectional links, all having the same propagation delay, equal to 1 ms. Links capacities are indicated in the figure, and the three highlighted links are the bottlenecks in this network topology. Twelve exponential On–Off traffic sources are considered, and their source and destination nodes are indicated in Fig. 9. Table 4 reports the peak rate, the subscribed rate, the utility function and the path for all the connections. Note that, with such paths choice, various connections compete for network capacity with different connections on bottleneck links. All other parameters are set as in the previous scenarios.

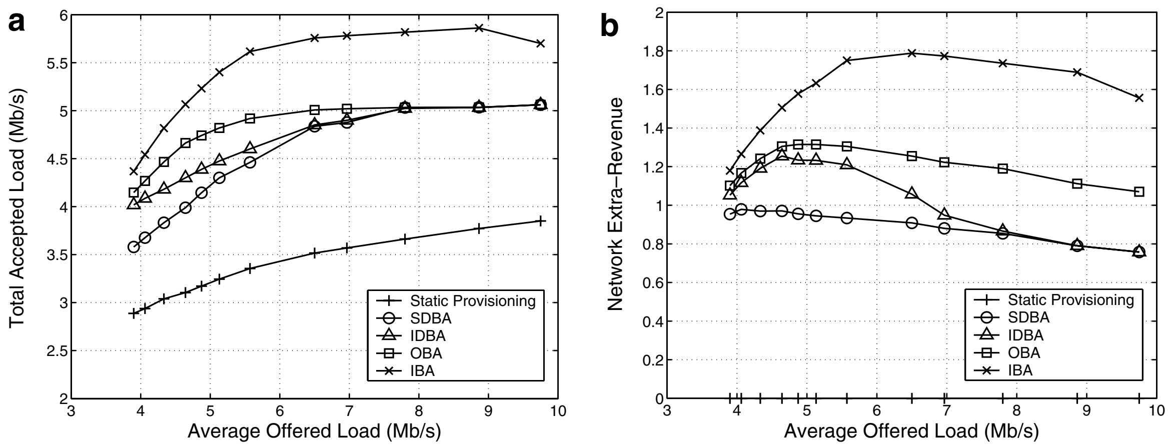

Fig. 8. Average total accepted load (a) and network extra-revenue (b) as a function of the average load offered to the network of Fig. 4. Nine aggregate traffic sources generate LRD traffic obtained multiplexing 10 Pareto On–Off sources. Three sources have utility function $U_1(x) = \frac{3}{2} \cdot \log(1 + x)$, while all the others have associated $U_2(x) = \frac{1}{2} \cdot \log(1 + x)$.

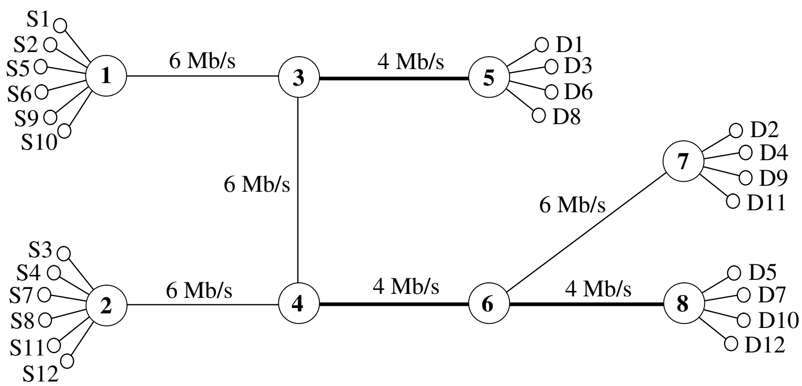

Fig. 9. Network topology with multiple bottleneck links.

Table 4
Peak rate, subscribed rate, utility function and path for the connections in the network scenario of Fig. 9

Connection	Peak rate (kb/s)	Subscribed rate (kb/s)	Utility function	Path
1	40	100	$0.5\log(1+x)$	1–3–5
2	40	100	$0.5\log(1+x)$	1–3–4–6–7
3	40	100	$0.5\log(1+x)$	2–4–3–5
4	40	100	$0.5\log(1+x)$	2–4–6–7
5	500	300	$0.5\log(1+x)$	1–3–4–6–8
6	500	300	$0.5\log(1+x)$	1–3–5
7	500	300	$0.5\log(1+x)$	2–4–6–8
8	500	300	$0.5\log(1+x)$	2–4–3–5
9	1000	300	$1.5\log(1+x)$	1–3–4–6–7
10	1000	300	$1.5\log(1+x)$	1–3–4–6–8
11	1000	300	$1.5\log(1+x)$	2–4–6–7
12	1000	300	$1.5\log(1+x)$	2–4–6–8

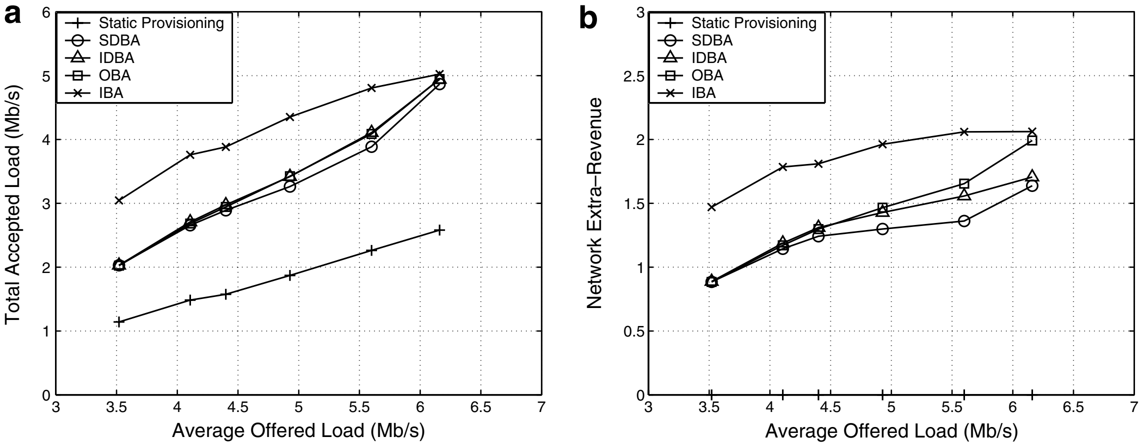

Fig. 10. Average total accepted load (a) and network extra-revenue (b) as a function of the average load offered to the network of Fig. 9.

Fig. 10 reports the total accepted load and network revenue in this scenario. Even though OBA and IDBA achieve the same total accepted load, it can be observed in Fig. 10b that OBA outperforms IDBA in network revenue, especially for high network loads, where it achieves almost 18% higher revenue than IDBA.

Finally, we considered the network topology proposed in [8] and illustrated in Fig. 11. This network scenario is more complex than the previous ones and it allows us to test our proposed allocation algorithms in a more realistic core network topology. It comprises 11 core nodes, 8 access nodes, 36 end nodes (18 source–destination pairs) and 28 bidirectional links, all having the same propagation delay, equal to 5 ms. The capacities of the links are indicated in the figure.

Eighteen exponential On–Off connections are offered to the network. Table 5 reports the peak rate, the subscribed rate, the utility function and the path for all the connections. The connections' paths are the same as in [8].

Fig. 12 shows the performance of the considered bandwidth allocation algorithms as a function of the total load offered to the network. The results are in line with those observed in the previous network scenarios and show that our proposed allocation algorithms allow to increase both total accepted traffic and network revenue with respect to a static allocation technique. Finally note that in this scenario the performance of IDBA almost overlaps that of OBA. This is expected since in this particular network topology, with the settings considered, few connections share capacity on each link and the utilization of enhanced allocation algorithms does not increase consistently network performance.

The proposed mathematical model, IBA, provides an upper bound on the performance achievable by any on-line allocation algorithm.

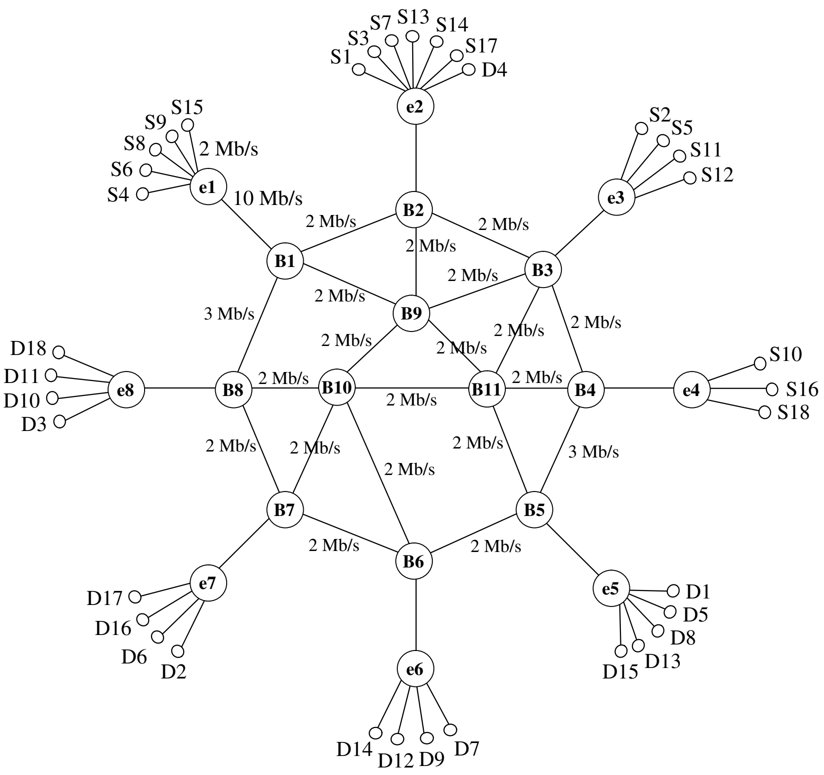

Fig. 11. Network topology with a larger number of links.

Table 5
Peak rate, subscribed rate, utility function and path for the connections in the network scenario of Fig. 11

Connection	Peak rate (kb/s)	Subscribed rate (kb/s)	Utility function	Path
1	100	300	$0.5\log(1+x)$	e2–B2–B3–B4–B5–e5
2	100	300	$0.5\log(1+x)$	e3–B3–B9–B10–B7–e7
3	100	300	$0.5\log(1+x)$	e2–B2–B1–B8–e8
4	100	300	$0.5\log(1+x)$	e1–B1–B2–e2
5	100	300	$0.5\log(1+x)$	e3–B3–B4–B5–e5
6	100	300	$0.5\log(1+x)$	e1–B1–B8–B7–e7
7	500	400	$0.5\log(1+x)$	e2–B2–B9–B10–B6–e6
8	500	400	$0.5\log(1+x)$	e1–B1–B9–B11–B5–e5
9	500	400	$0.5\log(1+x)$	e1–B1–B8–B7–B6–e6
10	500	400	$0.5\log(1+x)$	e4–B4–B11–B10–B8–e8
11	500	400	$0.5\log(1+x)$	e3–B3–B2–B1–B8–e8
12	500	400	$0.5\log(1+x)$	e3–B3–B4–B5–B6–e6
13	1000	500	$1.5\log(1+x)$	e2–B2–B3–B4–B5–e5
14	1000	500	$1.5\log(1+x)$	e2–B2–B9–B10–B6–e6
15	1000	500	$1.5\log(1+x)$	e1–B1–B9–B11–B5–e5
16	1000	500	$1.5\log(1+x)$	e4–B4–B5–B6–B7–e7
17	1000	500	$1.5\log(1+x)$	e2–B2–B1–B8–B7–e7
18	1000	500	$1.5\log(1+x)$	e4–B4–B11–B10–B8–e8

Two main factors impact on the performance of our proposed heuristic bandwidth allocation algorithms: traffic prediction and bandwidth allocation granularity. Let us recall that in the bandwidth allocation process, *non-greedy* sources are assigned a bandwidth that can accommodate traffic growth based on past traffic measurements. However, in this process the bandwidth allocated to a source

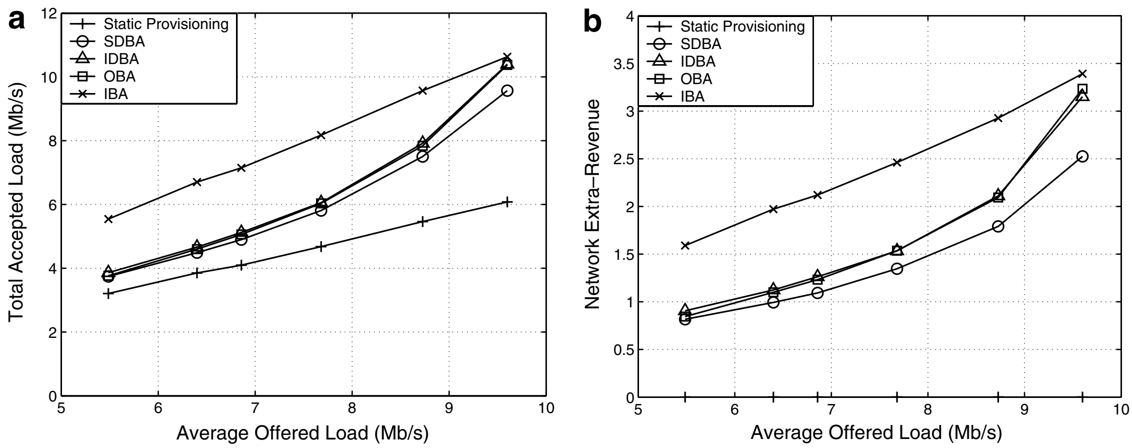

Fig. 12. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the complex core network of Fig. 11.

can exceed its actual transmission rate. At the same time, our allocation algorithms can assign excess bandwidth to *greedy* sources, thus leading to a potential bandwidth wastage.

On the other hand, since bandwidth allocation is performed only every T_u seconds, traffic variations can be tracked only with such granularity, leading again to potential inefficiency in bandwidth allocation.

The behavior observed in the network scenarios of Figs. 9 and 11 shows that the impact of traffic prediction is less remarkable with respect to that of the update interval T_u. In effect, when all exponential sources are always on and transmit at their peak rate (i.e. Off time = 0, the rightmost point in Figs. 10 and 12), no traffic variations occur at all, and T_u has no impact on the performance of the allocation algorithms. It can be observed that in this case the gap between IBA and the heuristic algorithms is negligible, meaning that the bandwidth wasted due to the inefficiency of the traffic predictor has little impact on the algorithms performance.

Since traffic prediction has even lower impact when sources are not always active (i.e. Off time $\geqslant 0$), we can conclude that, in these network scenarios, the gap between IBA and heuristic algorithms is mainly due to the bandwidth allocation granularity for every network load value.

On the other hand, in the single-bottleneck scenario of Fig. 4, the number of sources that share the bottleneck link is higher than in the other network scenarios, and the impact of traffic prediction is not anymore negligible. This can be observed in Figs. 6 and 7, where the gap between IBA and the heuristic allocation algorithms is evident for all offered loads.

Finally, we note that in all the considered network scenarios, the performance of the proposed dynamic bandwidth allocation algorithms (in particular, OBA) well approaches that of IBA, especially for high network loads.

To provide an upper bound on the signalling overhead of our proposed architecture, let us consider a network scenario with N_E ingress routers I_i, $i = 1,\ldots,N_E$, and M_i connections that access the network through I_i; let $M = \sum_{i=1}^{N_E} M_i$ be the total number of connections offered to the network. Every T_u seconds each ingress router exchanges update messages with all the other ingress routers to report the current statistics about incoming connections. For the most demanding allocation algorithm we proposed, OBA, the parameters that must be exchanged for each connection k are its transmission rate (b_k^{n-1}), its offered load (Ol_k^{n-1}), its subscribed rate (sr_k) and its utility function (U_k). If such information is encoded using b bits, then, every T_u seconds each ingress router I_i must send to all the other ingress routers $N_E - 1$ identical messages, each of length bM_i bits. The total number of bits transmitted by I_i is therefore equal to $(N_E - 1)bM_i$; hence a total of $B = \sum_{i=1}^{N_E}(N_E - 1)bM_i = (N_E - 1)bM$ signalling bits is generated each update interval. In the worst case, all the messages are transmitted over all the network links, so that the signalling bitrate occupied on each link will be equal to $\frac{B}{T_u} = \frac{(N_E-1)bM}{T_u}$ bit/s. However, signalling messages can be exchanged between the group of ingress routers using multicasting techniques, so that in the worst case only bM_i bits are transmitted by ingress router I_i on each link, and the overall signalling bitrate becomes equal to $\frac{bM}{T_u}$ bit/s. If we consider for example a network scenario with $N_E = 10$ ingress routers, $M = 1000$ connections, $T_u = 20$ s and $b = 64$ bits, we obtain a signalling overhead of 28.8 kb/s over each link in the worst case using unicast messages; if multicast messages are used the overhead is reduced to 3.2 kb/s.

Following the same reasoning we can set an upper bound on the delay involved in the exchange of signalling messages. Let D be the maximum network diameter (i.e. the maximum shortest path between any two nodes in the network), and C the capacity of the network bottleneck. If multicast messages are used, then bM bits must be transmitted. It is reasonable to expect that the multicast tree diameter is not much greater than D; hence, if we assume that signalling messages must traverse at most $2D$ links, $2D \cdot \frac{bM}{C}$ seconds are necessary in the worst case, supposing that propagation delays are negligible. In the same example network scenario used above, with $D = 10$ links and $C = 2$ Mb/s, it takes at most 0.64 s to transmit all signalling messages among ingress routers.

Based on the numerical example illustrated above we believe that the signalling overhead and delay are largely acceptable in most practical scenarios. However, note that both overhead and delay depend on the total number of connections offered to the network, M, and their impact on system behavior can be considered negligible only as long as M is below a threshold which is obviously related to the number of ingress routers and the network size.

8. Conclusion

In this paper we proposed a novel service model where users subscribe for guaranteed transmission rates, and the network periodically individuates unused bandwidth that is re-allocated and guaranteed with short-term contracts to users who can better exploit it according to their traffic needs and bandwidth utility functions.

We described a distributed dynamic resource provisioning architecture for quality of service networks. We developed a set of efficient bandwidth allocation algorithms that take explicitly into account traffic statistics to increase the users perceived utility and the network revenue. Our proposed algorithms allow to overcome static allocation and overbooking techniques, since they perform bandwidth allocation adaptively, maximizing users' satisfaction and allowing higher resource utilization and network service differentiation.

We developed a mathematical model that allows an optimal bandwidth allocation from the provider point of view, maximizing network revenue. This model provides upper bounds to the performance achievable by on-line dynamic bandwidth allocation algorithms.

Simulation results measured in realistic network scenarios showed that our allocation algorithms and service model increase consistently both resource utilization and network revenue with respect to static provisioning techniques.

Further, we showed that our bandwidth allocation algorithms approach the upper bounds provided by the mathematical model in the considered network scenarios.

References

[1] A. Capone, J. Elias, F. Martignon, G. Pujolle, Dynamic resource allocation in communication networks, in: Proceedings of Networking 2006, Coimbra, Portugal, 15–19 May 2006, also published in Springer LNCS, vol. 3976, 2006, pp. 892–903.

[2] A.T. Campbell, R.R.-F. Liao, Dynamic core provisioning for quantitative differentiated services, IEEE/ACM Transactions on Networking 12 (3) (2004) 429–442.

[3] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, J.E. van der Merwe, Resource management with hoses: point-to-cloud services for virtual private networks, IEEE/ACM Transactions on Networking 10 (5) (2002) 679–692.

[4] D. Bertsekas, R. Gallager, Data Networks, second ed., Prentice-Hall, 1992.

[5] M. Allalouf, Y. Shavitt. Centralized and distributed approximation algorithms for routing and weighted max–min fair bandwidth allocation, in: IEEE Workshop on High Performance Switching and Routing (HPSR'05), China, May 2005.

[6] Z. Cao, E. Zegura. Utility max–min: an application-oriented bandwidth allocation scheme, in: Proceedings of the IEEE Infocom'99, New York, USA, March 1999.

[7] F.P. Kelly, Charging and rate control for elastic traffic, European Transactions on Telecommunications 8 (1) (1997) 33–37.

[8] R.J. La, V. Anantharam, Utility-Based rate control in the internet for elastic traffic, IEEE/ACM Transactions on Networking 10 (2) (2002) 272–286.

[9] J.-W. Lee, R.R. Mazumdar, N.B. Shroff, Non-Convex optimization and rate control for multi-class services in the internet, IEEE/ACM Transactions on Networking 13 (4) (2005) 827–840.

[10] C. Touati, E. Altman, J. Galtier, Generalized nash bargaining solution for bandwidth allocation, Computer Networks 50 (17) (2006) 3242–3263.

[11] M. Chiang, S. Zhang, P. Hande, Distributed rate allocation for inelastic flows: optimization frameworks, optimality

conditions, and optimal algorithms, in: Proceedings of the IEEE Infocom'05, vol. 4, Miami, USA, 13–17 March 2005, pp. 2679–2690.
[12] R.M. Salles, J.A. Barria, Fair and efficient dynamic bandwidth allocation for multi-application networks, Computer Networks 49 (6) (2005) 856–877.
[13] A.T. Campbell, R.R.-F. Liao, Dynamic edge provisioning for core IP networks. in: Proceedings of the IEEE/IFIP International Workshop on Quality of Service IWQOS, Pittsburgh, USA, June 2000.
[14] T. Ahmed, R. Boutaba, A. Mehaoua, A measurement-based approach for dynamic QoS adaptation in DiffServ networks, Journal of Computer Communications (2004) (special issue on End-to-End Quality of Service Differentiation).
[15] E.W. Fulp, D.S. Reeves, Bandwidth provisioning and pricing for networks with multiple classes of service, Computer Networks 46 (1) (2004) 41–52.
[16] H. Schulzrinne, X. Wang, Incentive-compatible adaptation of internet real-time multimedia, IEEE Journal on Selected Areas in Communications 23 (2) (2005) 417–436.
[17] M. Mahajan, M. Parashar, A. Ramanathan, Active resource management for the differentiated services environment, International Journal of Network Management 14 (3) (2004) 149–165.
[18] H.T. Tran, T. Ziegler, Adaptive bandwidth provisioning with explicit respect to QoS requirements, Computer Communications 28 (16) (2005) 1862–1876.
[19] J. Aweya, M. Ouellette, D.Y. Montuno, Design and stability analysis of a rate control algorithm using the Routh-Hurwitz stability criterion, IEEE/ACM Transactions on Networking 12 (4) (2004) 719–732.
[20] J. Aweya, M. Ouellette, D.Y. Montuno, A simple, scalable and provably stable explicit rate computation scheme for flow control in computer networks, International Journal of Communication Systems 14 (6) (2001) 593–618.
[21] S. Shenker, Fundamental design issues for the future internet, IEEE Journal on Selected Areas in Communications 13 (7) (1995) 1176–1188.
[22] R. Jain, S. Kalyanaraman, S. Fahmy, R. Goyal, Seong-Cheol Kim, Source behavior for ATM ABR traffic management: an explanation, IEEE Communications Magazine 34 (11) (1996) 50–55.
[23] K. Ramakrishnan, S. Floyd, D. Black, The addition of explicit congestion notification (ECN) to IP, in: IETF RFC 3168, September 2001.
[24] M. Minoux, Mathematical Programming: Theory and Algorithms, Wiley, 1986.
[25] E. Rosen, A.Viswanathan, R. Callon, Multiprotocol label switching architecture, in: IETF RFC 3031, January 2001.
[26] Generalized multi-protocol label switching (GMPLS) signaling functional description, in: L. Berger (Ed.), IETF RFC 3471, January 2003.
[27] A. Banerjee, L. Drake, L. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, Y. Rekhter, Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques, IEEE Communications Magazine 39 (7) (2001) 144–151.
[28] A.K. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the single-node case, IEEE/ACM Transactions on Networking 1 (3) (1993) 344–357.
[29] H. Schulzrinne, X. Wang, RNAP: A resource negotiation and pricing protocol, in: International workshop on network and operating system support for digital audio and video (NOSSDAV), Basking Ridge, NJ, June 1999, pp. 77–93.
[30] J-Sim, Ohio State University. Available at www.j-sim.org.
[31] K. Pawlikowski, H.-D. Joshua Jeong, J.-S. Ruth Lee, On credibility of simulation studies of telecommunications networks, IEEE Communications Magazine (2002) 132–139.
[32] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson, Self-Similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level, IEEE/ACM Transactions on Networking 5 (1) (1997) 71–86.

Jocelyne Elias received her Master of Computer Sciences and Telecommunications Engineering from the Lebanese University of Tripoli in 2002, the Master's Degree (DEA) in Advanced Networks of Knowledge and Organization from University of Technology of Troyes in 2003, and the Ph.D. degree in Computer Science from University of Pierre et Marie Curie, Paris, in July 2006. She is now a Post-doc researcher at LIP6 Laboratory, Paris. Her current research activities include dynamic resource allocation in quality of service networks.

Fabio Martignon received the Laurea and the Ph.D. degree in telecommunication engineering from the Politecnico di Milano in October 2001 and May 2005, respectively. He is now an assistant professor in the Department of Management and Information Technology at the University of Bergamo. His current research activities include routing for multihop wireless networks, congestion control and QoS routing over IP networks.

Antonio Capone received the Laurea degree (MS degree equivalent) and the Ph.D. degree in telecommunication engineering from the Politecnico di Milano in July 1994 and June 1998, respectively. In 2000, he was a visiting scientist at the University of California, Los Angeles. He is now an associate professor in the Department of Electronics and Information at the Politecnico di Milano. His current research activities include packet access in wireless cellular networks, routing and MAC for multihop wireless networks, congestion control and QoS issues of IP networks, network planning and optimization. He is a member of the IEEE and the IEEE Communications and Vehicular Technology Societies.

Guy Pujolle is currently a Professor at the University of Paris VI, Paris, France, and a member of the Scientific Council of France Telecom Group. He is Chairman of IFIP Working Group 6.2 on Network and Internetwork Architectures. He is a pioneer in high-speed networking having led the development of the first gigabit/s network to be tested in 1980. He is an Editor for the International Journal of Network Management, ACM Wireless Networks, Telecommunication Systems, and Annals of Telecommunications. He is Co-Founder and Member of the Scientific Council of QoSMOS (www.QoSMoS.fr), Ucopia Communications, Inc. (www.ucopia.com), and Ginkgo-Networks (www.ginkgo-networks.com). He is an Editor for the IEEE Tutorial and Survey.