Distributed Algorithms for Dynamic Bandwidth Provisioning in Communication Networks

Jocelyne Elias[1], Fabio Martignon[2], Antonio Capone[3], Guy Pujolle[1]

[1] University of Paris 6, LIP6 Laboratory, Paris, France
Email: {jocelyne.elias,guy.pujolle}@lip6.fr
[2] Department of Management and Information Technology, University of Bergamo, Italy
Email: fabio.martignon@unibg.it
[3] Department of Electronics and Information of Politecnico di Milano, Italy
Email: capone@elet.polimi.it

***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 quantitative 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 an efficient bandwidth allocation algorithm that takes explicitly into account traffic statistics to increase the users' benefit and the network revenue simultaneously. We demonstrate through simulation in several realistic network scenarios that the proposed dynamic provisioning model is superior to static provisioning in providing resource allocation both in terms of total accepted load and network revenue.**

***Index Terms* - Dynamic Bandwidth Allocation, Autonomic Networks, Service Model.**

I. Introduction

Efficient dynamic resource provisioning mechanisms are necessary to the development and automation of Quality of Service networks. In telecommunication 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 [1].

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 [1-11]. These approaches and related works are discussed in Section II.

In this paper we 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 can better exploit it to transmit at a rate higher than their subscribed rate.

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 using signalling messages like those defined in [2]. 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 [3], [4], as well as of their actual sending rate. Based on such information and that communicated by core routers, ingress routers allocate network resources dynamically and efficiently using a modified version of the max-min fair allocation algorithm proposed in [5]. Such allocation is performed taking into account users' profile and willingness to acquire extra-bandwidth based on their bandwidth utility function. The allocation is then enforced by traffic conditioners that perform traffic policing and shaping.

We evaluate by simulation the performance of our proposed bandwidth allocation algorithm in realistic network scenarios. Numerical results show that our architecture allows 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 and the proposition of a distributed architecture that performs

This paper is based on "Dynamic Resource Allocation in Communication Networks", by A. Capone, J. Elias, F. Martignon, and G. Pujolle, which appeared in the Proceedings of the Networking 2006 Conference, Coimbra, Portugal, May 2006. © 2006 Springer.

dynamic bandwidth allocation to maximize users' utility and network revenue.

The paper is structured as follows: Section II discusses related work; Section III introduces our proposed service model and provisioning architecture; Section IV describes the proposed dynamic bandwidth allocation algorithm and Section V discusses its convergence property; Section VI presents simulation results that show the efficiency of our dynamic resource allocation algorithm compared to a static allocation technique. Finally, Section VII concludes this work and outlines future research issues.

II. Related Work

The problem of bandwidth allocation in telecommunication networks has been addressed in many recent works. Both allocation algorithms [5], [6] and provisioning architectures [1], [2], [7], [8], [9], [10], [11] have been proposed in the literature, and are reviewed in the following.

In [5] a max-min fair allocation algorithm is proposed to allocate bandwidth equally among all connections bottlenecked at the same link. The authors in [6] extend the max-min fair allocation algorithm to the case where each flow may 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 [5] to perform a periodic allocation of unused bandwidth, through short-term contracts, to users who are willing to transmit more than their subscribed rate.

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 [1], [2], [7], [8], [9], [10], [11].

In [1], [7], the authors propose a dynamic core and edge provisioning architecture for differentiated services IP networks. The basic role of dynamic edge provisioning is to perform dynamic ingress link sharing and dynamic egress capacity dimensioning. The core provisioning architecture consists of a set of dynamic node and core provisioning algorithms for interior nodes and core networks, respectively. The node provisioning algorithm adopts a self-adaptive mechanism to adjust service weights of weighted fair queuing schedulers at core routers while the core provisioning algorithm reduces edge bandwidth immediately after receiving a Congestion-Alarm signal from a node provisioning module and provides periodic bandwidth re-alignment to establish a modified max-min bandwidth allocation to traffic aggregates.

The work discussed in [1] has similar objectives to our dynamic bandwidth allocation algorithm. However, their service model differs from our proposed model and traffic statistics are not taken into account in the allocation procedure. Moreover, in our work we suggest a distributed architecture implementation, while in these papers only a centralized scheme is considered.

A policy-based architecture is presented in [8], where a measurement-based approach is proposed for dynamic Quality of Service adaptation in DiffServ networks. The proposed architecture is composed of one Policy Decision Point (PDP), a set of Policy Enforcement Points that are installed in ingress routers and bandwidth monitors implemented in core routers. When monitors detect significant changes in available bandwidth they inform the PDP which changes dynamically the policies on in-profile and out-of-profile input traffics based on the current state of the network estimated using the information collected by the 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 [9], the authors address the problem of bandwidth provisioning and pricing for networks with multiple classes of service. A connection management strategy for QoS enabled networks is introduced to maximize service providers revenue, while reducing blocking experienced by users. Moreover, the authors address and analyze the issues regarding demand estimation, connection duration, and pricing intervals in the connection management framework.

In [2], 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 [2], we take into account users bandwidth utility functions to evaluate our proposed allocation algorithm based on the increased network revenue that is achieved. However, the authors 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 [3], [4], [10]. In [10], the authors propose an active resource management approach (ARM) for 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 reallocated without loss of service. This concept is in line with our proposed bandwidth allocation algorithm. 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.

III. Service Model and Dynamic Provisioning Architecture

We first introduce our proposed service model, then we present a distributed provisioning architecture which implements such service model by performing the dynamic bandwidth allocation algorithm described in Section IV; finally, we present the signalling messages used to assure the interaction between network elements.

A. Service Model

We propose a service model that, first, provides a quantitative bandwidth guarantee to users and then exploits the unused bandwidth individuated periodically in the network to propose short-term guaranteed extra-bandwidth. In this process, different weights can be assigned to network users to allocate extra-bandwidth with different priorities; such weights can be set statically offline, 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 with guarantee, during the update interval, to users who can exploit it to transmit extra-traffic;
- a weight that expresses the user's priority in the assignment of extra-bandwidth;
- a bandwidth utility function, $U(x)$, that describes the user's preference for an allocation of x bandwidth units. In line with [12] we consider the utility function as part of the service model. Without loss of generality, we do not consider the pricing component of a bandwidth utility function.

B. Architecture and Control Messaging

To implement our service model we assume a distributed architecture constituted by core and edge routers, as shown in Figure 1; traffic monitors are installed on ingress and core routers to perform on-line measurements on the incoming traffic flows and network capacity utilization, respectively. Core routers exchange messages with ingress routers to report the link utilization or to notify a congestion situation. Each ingress router collects the measurements performed by traffic monitors and exchanges periodically update messages with all other ingress routers to report the current incoming traffic statistics. Moreover, a dynamic bandwidth allocation algorithm is implemented in all ingress routers: it takes into account the traffic statistics gathered at ingress routers and the network information reported by core routers to allocate network resources dynamically and efficiently.

The messages exchanged between network routers, illustrated with arrows in Figure 1, are similar to the control messages proposed in [1] to report persistent congestion or resource availability. A subset of the messages defined in the RNAP protocol [13] can be used for these purposes.

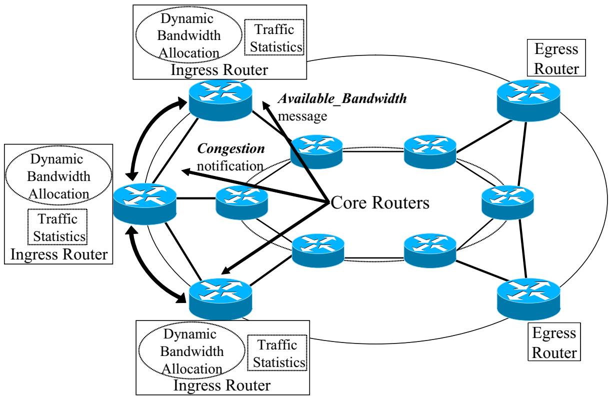


Figure 1. The proposed distributed architecture that supports dynamic bandwidth allocation

Since dynamic provisioning algorithms are complementary to admission control algorithms [1], 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.

IV. Dynamic Bandwidth Allocation Algorithm

We propose a novel dynamic provisioning algorithm, named Simple Dynamic Bandwidth Allocation algorithm (SDBA), that allocates network capacity efficiently based on traffic statistics measured on-line. Bandwidth allocation is performed by ingress routers periodically and is enforced using traffic conditioners. We denote the interval between two successive allocations performed by the algorithm as the *update interval*, whose duration is T_u seconds. Moreover, core routers monitor link utilization, and if congestion on some links is detected, bandwidth reallocation is immediately invoked to solve this situation.

In the following we present in details the SDBA algorithm, that proceeds in two steps: in the first step, bandwidth is allocated to all active connections trying to match their near-term traffic requirements that are predicted based on statistics collected by ingress routers. In step two, spare bandwidth as well as 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 already fully exploiting their subscribed rate.

To illustrate SDBA, 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), for $k = 1, \ldots, K$, where s_k, d_k and sr_k represent the connections source node, destination node and the subscribed rate, respectively; furthermore, we assume that each connection has associated r_min_k, which represents the minimum bandwidth the application requires. 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.

At the beginning of the $n-th$ update interval, each ingress router computes the transmission rate, b_k^{n-1}, averaged over the last T_u seconds, for all connections $k \in K$ that access the network through it. This information is then sent to all other ingress routers using control messages as described in the previous Section, so that all ingress routers can share the same information about current traffic statistics and perform simultaneously the same allocation procedure.

The amount of bandwidth allocated to each source k during the $n-th$ update interval, r_k^n, is determined using the two-steps approach described in the following:

- First step: Connections having $b_k^{n-1} < r\_min_k$ are considered $idle$; all other active connections are further classified as $greedy$ if they used a fraction greater than γ of their subscribed rate sr_k (i.e. if $b_k^{n-1} > \gamma \cdot sr_k$), otherwise they are classified as $non-greedy$.
  Let us denote by K_i, K_{ng} and K_g the sets of idle, non-greedy and greedy connections, respectively.
  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. In this work we only consider the last measured value, b_k^{n-1}, and we propose the following simple bandwidth allocation: $r_k^n = \min\{2 \cdot b_k^{n-1}, sr_k\}, \forall k \in K_{ng}$. In this regard we are currently studying more efficient traffic predictors that could allow improved bandwidth allocation.
  Greedy connections are assigned in this step their subscribed rate sr_k, and they also take part to the allocation of extra-bandwidth performed in step two, since they are already exploiting all their subscribed rate.
- Second step: after having performed the allocations described in step one, the 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:

$$R_l = C_l - \Big(\sum_{k \in K_i \cup K_{ng}} r_k^n \cdot a_k^l + \sum_{k \in K_g} sr_k \cdot a_k^l \Big), \forall l \in L \quad (1)$$

where the first summation represents the total bandwidth allocated in step one 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 connections using the algorithm detailed in Table I, which is an extension of the allocation algorithm proposed in [5]. 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 $n-th$ update interval, so that finally $r_k^n = sr_k + f_k^n, \forall k \in K_g$.

TABLE I.
PSEUDO-CODE SPECIFICATION OF THE SIMPLE DYNAMIC BANDWIDTH ALLOCATION ALGORITHM (SDBA)

(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 $l \in L$, calculate $F_l = R_l / n_l$
(4)	identify the link α that minimizes F_α i.e. $\alpha \mid F_\alpha = min_k(F_k)$
(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)

To illustrate the operation of SDBA, let us refer to the simple network scenario shown in Figure 2, where four greedy connections are active in the $n-th$ update interval: two connections (C_1 and C_2) between nodes (A,C) and two connections (C_3 and C_4) between nodes (B,C). Connections paths are reported in Table II, while the residual capacity R_l, expressed in bandwidth units, is indicated for each link in Figure 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 $f_{C_1}^n = f_{C_2}^n = 0.5$. 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 $f_{C_3}^n = f_{C_4}^n = 1$.

To take into account users weights it is sufficient to substitute n_l in Table I with w_l, which is defined as the sum of the weights of all greedy connections that are

TABLE II.
PATH FOR THE CONNECTIONS IN THE EXAMPLE SCENARIO OF FIGURE 2

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

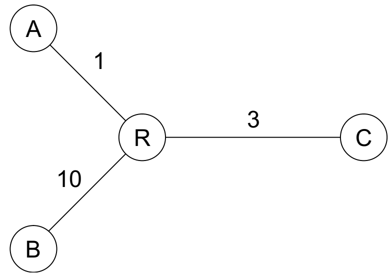


Figure 2. Example scenario that illustrates the operation of the SDBA algorithm: two connections are established between nodes (A,C) and between nodes (B,C)

routed on link l.

It should be clarified that our algorithm 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 Section VI 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.

V. CONVERGENCE PROPERTY

We briefly discuss the convergence property of our proposed allocation algorithm, following closely the analysis proposed in [14]. It has been shown that a distributed algorithm needs at least M iterations to stabilize toward max-min allocation in a descending order starting from the most congested bottleneck link, where M is the number of distinct bottleneck links in the network.

Since SDBA is an extended version of the max-min fair share algorithm introduced in [5], its convergence speed depends on the set of bottleneck links and how connections are routed in the network sharing such bottleneck links. Various traffic sources can send traffic over the same congested links, a situation that arises frequently in communication networks. In the extreme case, when all the sources have portions of traffic over all the congested links, these sources are only constrained by the most congested bottleneck link. In this case, our algorithm takes one round to finish, and the allocation is done with respect to the capacity of the most congested bottleneck link.

VI. NUMERICAL RESULTS

In this Section we compare the performance, measured by the average accepted load and network extra-revenue versus the total load offered to the network, of the proposed dynamic bandwidth allocation algorithm (SDBA) with a static provisioning strategy, referring to different network scenarios to cover a wide range of possible environments. Static provisioning allocates to each source k its subscribed rate sr_k.

We are interested in measuring 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.

We define, in line with [2], the average network extra-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 extra-revenue generated by greedy users that are assigned extra-bandwidth by our proposed dynamic allocation algorithm. Furthermore we assume, in line with [15], 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 Section IV, the average network extra-revenue can be obtained averaging over all the update intervals n the quantity:

$$\sum_{k \in K_g} U(b_k^n) - U(sr_k) \qquad (2)$$

All numerical results have been calculated over long-lived data exchanges, achieving very narrow 95% confidence intervals [16].

In the first scenario we gauge the effectiveness of the proposed traffic-based bandwidth allocation algorithm. We consider, in line with [1], [2], the scenario illustrated in Figure 3, that consists of a single-bottleneck with 2 core nodes, 6 access nodes, 40 end nodes (20 source-destination pairs) and traffic conditioners at the edge. Each ingress conditioner is configured with one profile for each traffic source, and drops out-of-profile packets. All links are full-duplex and have a propagation delay of 1 ms. 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.

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 to 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 constant 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, 8 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 and γ is set to 0.9.

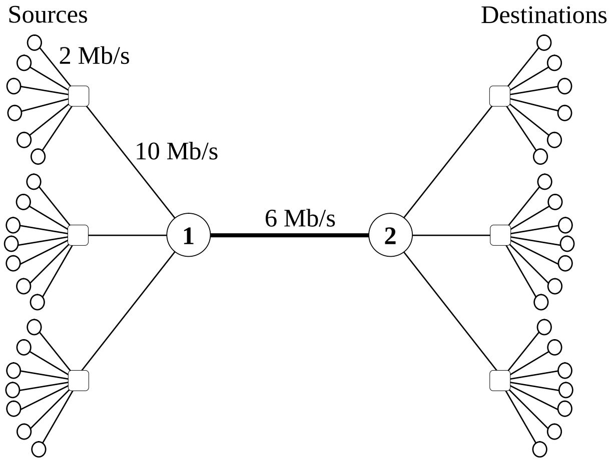
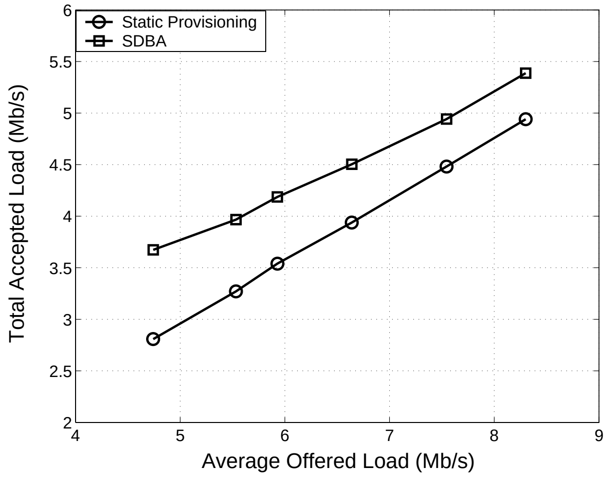
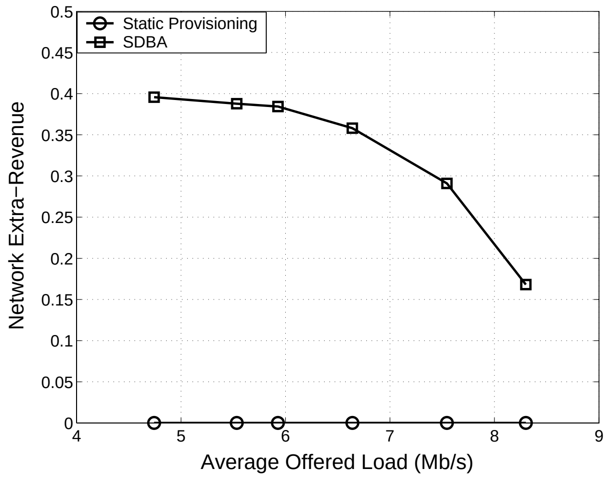


Figure 3. Network topology with a single bottleneck



(a)



(b)

Figure 4. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Figure 3

We assume, for simplicity, that all users have the same weight w_k and the same utility function proposed in [17], $U_k(x) = 0.5 \cdot log(1+x)$, that models the perceived utility of elastic traffic for an allocation of x bandwidth units.

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 algorithm.

Figures 4(a) and 4(b) show, 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.

It can be observed that SDBA is very efficient in resource allocation compared to a static provisioning algorithm for all values of the offered load, providing improvements up to 31% in the total accepted traffic.

The maximum network extra-revenue is achieved when the average Off time of Exponential sources is equal to 150 s, corresponding to an offered load approximately equal to 5 Mb/s. In this situation, in fact, the average number of idle connections (i.e. 9) is sufficiently high to exalt our dynamic allocation algorithm that re-allocates 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 on the performance of the update interval duration, 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 9% for $T_u = 40$ s and 7% for $T_u = 60$ s, while for $T_u = 20$ s it was 16% (see Figure 4(a)). These results allow to gauge the trade-off between performance improvement and overhead resulting from a more frequent execution of the allocation algorithm.

In the same scenario of Figure 3 we then 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 them by a factor α, with $0.25 \leq \alpha \leq 1.5$. Figures 5(a) and 5(b) show 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 provisioning. 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 needs 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 algorithm is evident both in terms of accepted load and network extra-revenue.

A more realistic scenario is shown in Figure 6. It comprises 6 nodes and 8 bidirectional links, all having a capacity equal to 2 Mb/s and propagation delay of 1 ms. In this topology, 6 Exponential On-Off traffic sources are considered, and their source and destination nodes are

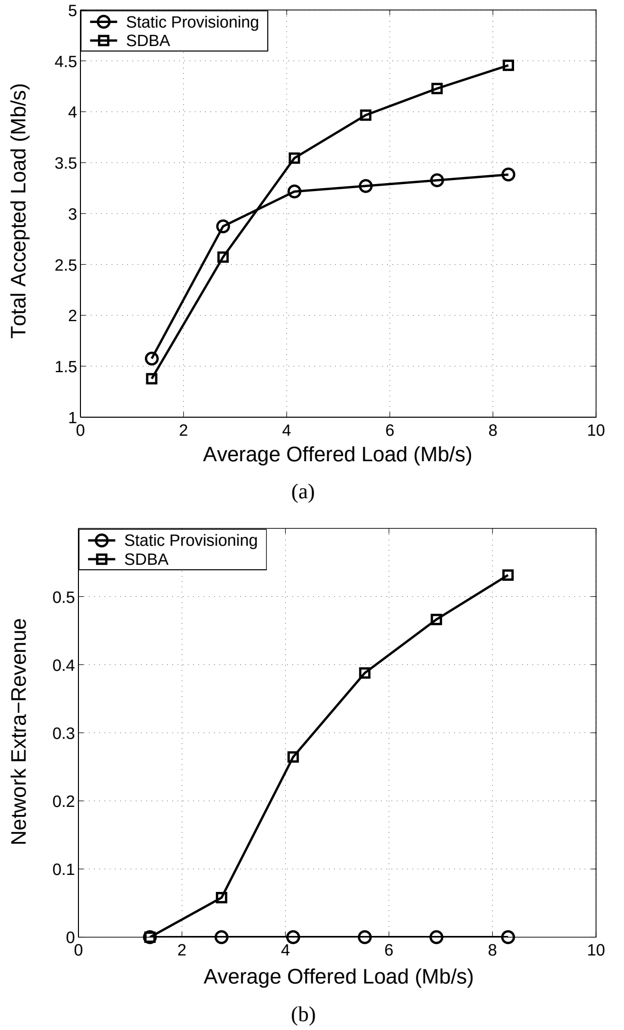


Figure 5. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Figure 3

indicated in the Figure. Table III reports the peak rate, the subscribed rate and the path for all the connections. All other parameters are set as in the previous scenarios. Note that, with such paths choice, various connections compete for network capacity with different connections on different links.

Also in this scenario SDBA outperforms static allocation, as shown in Figures 7(a) and 7(b), thus proving the benefit of the proposed scheme. These results verify that our allocation algorithm allows service providers to increase network capacity utilization and consequently network extra-revenue with respect to static provisioning techniques.

We then considered the network topology shown in Figure 8, originally proposed in [1]. It comprises 8 core nodes and 7 bidirectional links, all having the same propagation delay, equal to 1 ms. The capacities are given next to the links 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 Figure 8. Table IV reports the peak rate, the subscribed

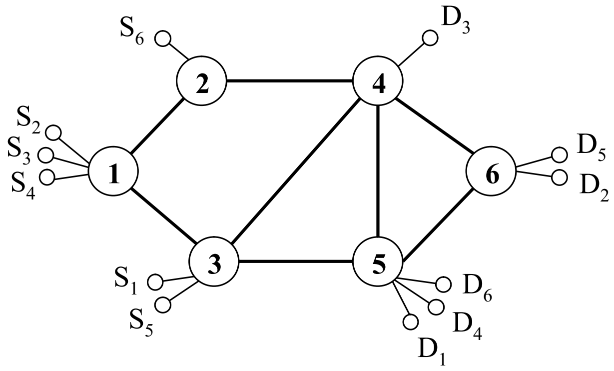


Figure 6. Network topology with a larger number of links

TABLE III.
PEAK RATE, SUBSCRIBED RATE AND PATH FOR THE CONNECTIONS IN THE NETWORK SCENARIO OF FIGURE 6

Connection	Peak Rate (kb/s)	Subscribed Rate (kb/s)	Path
1	100	300	3-4-5
2	500	400	1-2-4-6
3	500	400	1-3-4
4	1000	400	1-3-5
5	1000	400	3-4-6
6	1000	400	2-4-5

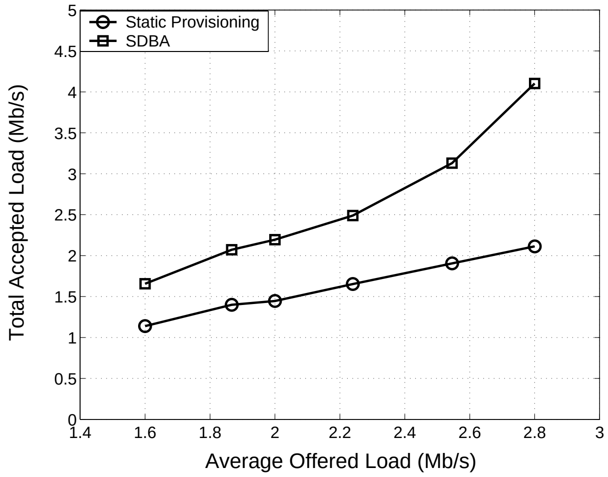
rate and the path for all the connections. All other parameters are set as in the previous scenarios. Also in this scenario, various connections compete for network capacity with different connections on different links.

It can be observed in Figures 9(a) and 9(b) that SDBA outperforms static provisioning both in terms of total accepted load and network revenue, especially for high network loads, where it achieves in average almost a 80% higher total accepted traffic.

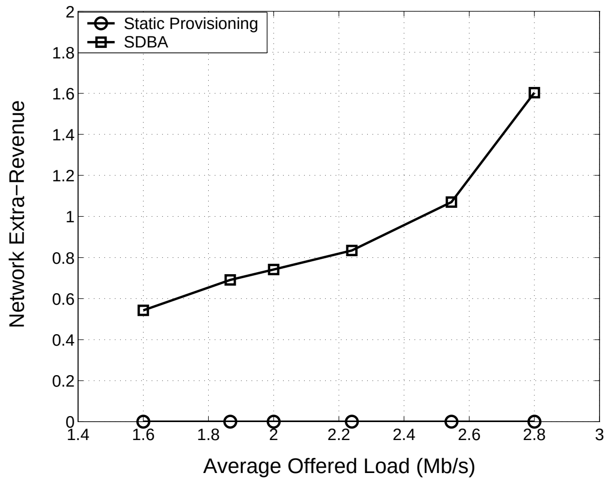
TABLE IV.
PEAK RATE, SUBSCRIBED RATE AND PATH FOR THE CONNECTIONS IN THE NETWORK SCENARIO OF FIGURE 8

Connection	Peak Rate (kb/s)	Subscribed Rate (kb/s)	Path
1	40	100	1-3-5
2	40	100	1-3-4-6-7
3	40	100	2-4-3-5
4	40	100	2-4-6-7
5	500	300	1-3-4-6-8
6	500	300	1-3-5
7	500	300	2-4-6-8
8	500	300	2-4-3-5
9	1000	300	1-3-4-6-7
10	1000	300	1-3-4-6-8
11	1000	300	2-4-6-7
12	1000	300	2-4-6-8

Finally we considered the network topology proposed in [17] and illustrated in Figure 10. This network scenario is more complex than the previous ones and it allows to test our proposed allocation algorithm 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



(a)



(b)

Figure 7. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Figure 6

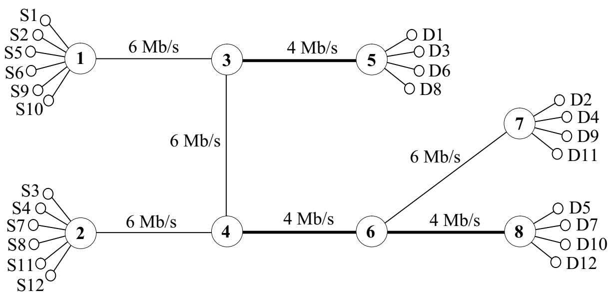
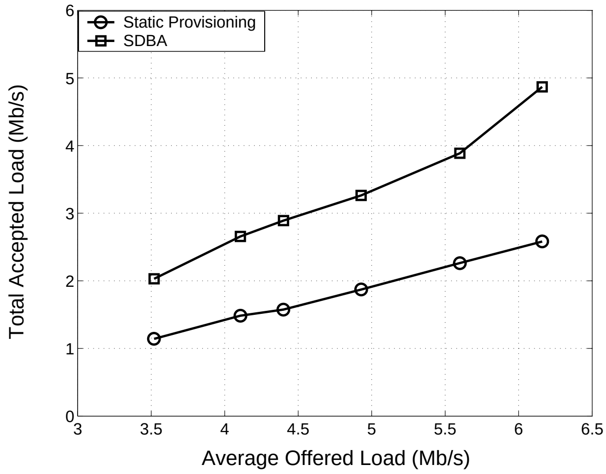


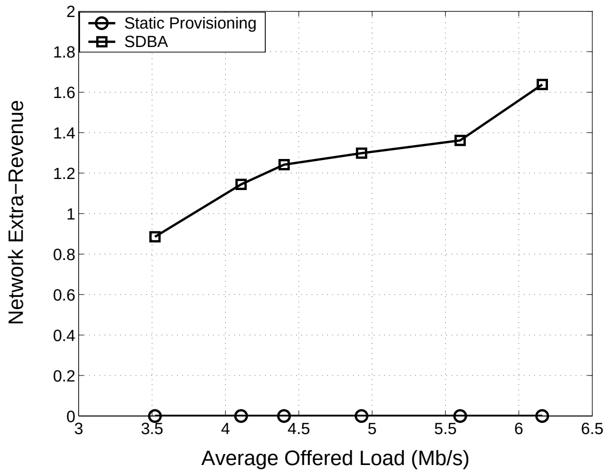
Figure 8. Network topology with multiple bottleneck links

delay, equal to 5 ms. The capacities are given next to the links in the Figure. Eighteen Exponential On-Off connections share the network. Table V reports the peak rate, the subscribed rate and the path for all the connections, which are the same as in [17].

Figures 11(a) and 11(b) show the performance of the considered bandwidth allocation algorithm as a function of the total load offered to the network. The results are in line with those achieved with the previous scenarios



(a)



(b)

Figure 9. Average total accepted load (a) and network extra-revenue (b) as a function of the average load offered to the network of Figure 8

and show that our proposed allocation algorithm allows to increase both total accepted traffic and network revenue with respect to a static allocation technique.

VII. 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. We described a distributed dynamic resource provisioning architecture for quality of service networks. We developed an efficient dynamic bandwidth allocation algorithm that takes explicitly into account traffic statistics to increase the users perceived utility and the network extra-revenue.

Simulations results measured in realistic network scenarios show that our allocation algorithm allows to increase both resource utilization and network revenue with respect to static provisioning techniques.

As a part of the future work, we plan to study more efficient bandwidth allocation algorithms that take into account the load offered to the network by each connection.

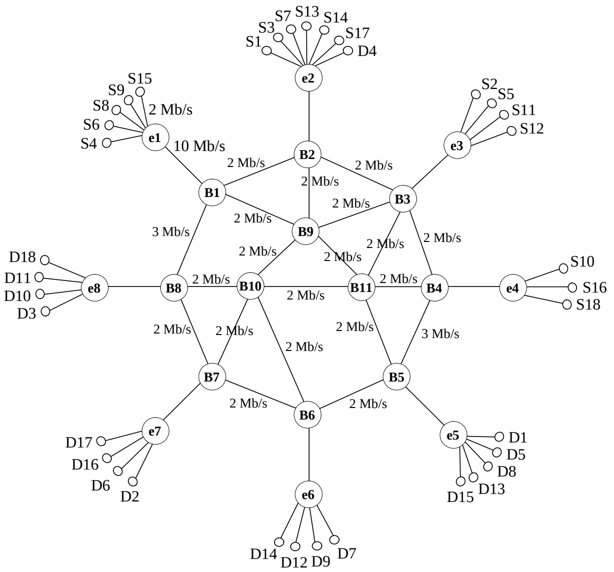


Figure 10. Complex core network topology

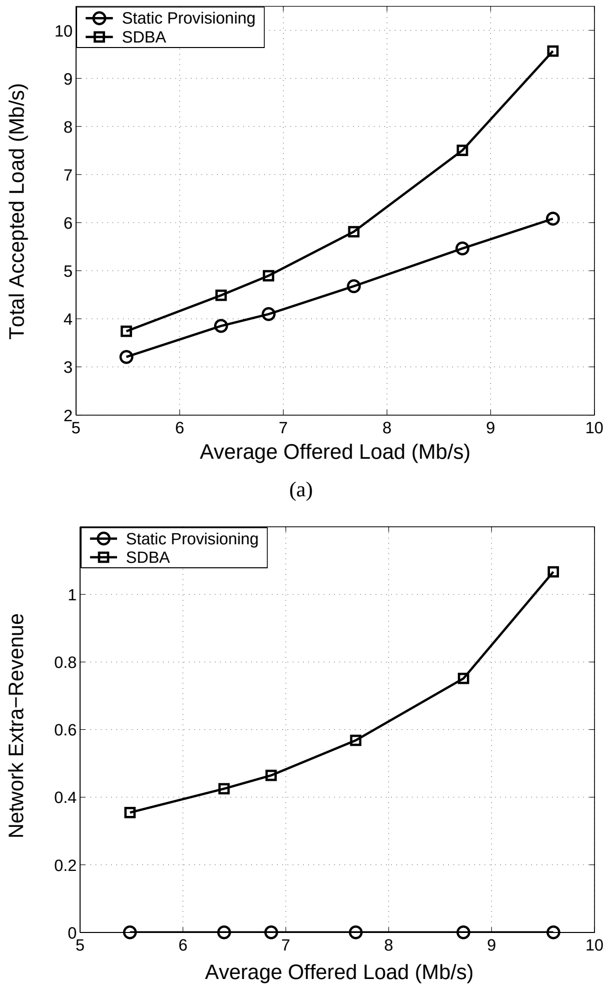
TABLE V.
PEAK RATE, SUBSCRIBED RATE AND PATH FOR THE CONNECTIONS IN THE NETWORK SCENARIO OF FIGURE 10

Connection	Peak Rate (kb/s)	Subscribed Rate (kb/s)	Path
1	100	300	e2-B2-B3-B4-B5-e5
2	100	300	e3-B3-B9-B10-B7-e7
3	100	300	e2-B2-B1-B8-e8
4	100	300	e1-B1-B2-e2
5	100	300	e3-B3-B4-B5-e5
6	100	300	e1-B1-B8-B7-e7
7	500	400	e2-B2-B9-B10-B6-e6
8	500	400	e1-B1-B9-B11-B5-e5
9	500	400	e1-B1-B8-B7-B6-e6
10	500	400	e4-B4-B11-B10-B8-e8
11	500	400	e3-B3-B2-B1-B8-e8
12	500	400	e3-B3-B4-B5-B6-e6
13	1000	500	e2-B2-B3-B4-B5-e5
14	1000	500	e2-B2-B9-B10-B6-e6
15	1000	500	e1-B1-B9-B11-B5-e5
16	1000	500	e4-B4-B5-B6-B7-e7
17	1000	500	e2-B2-B1-B8-B7-e7
18	1000	500	e4-B4-B11-B10-B8-e8

Further, we plan to develop a mathematical model of the bandwidth allocation problem that will provide theoretical bounds to the performance achieved by on-line allocation schemes.

ACKNOWLEDGEMENTS

This work is partially supported by the National Council for Scientific Research in Lebanon.



(b)

Figure 11. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the complex core network of Figure 10

REFERENCES

[1] A. T. Campbell and R. R.-F. Liao, "Dynamic Core Provisioning for Quantitative Differentiated Services," *IEEE/ACM Transactions on Networking*, pp. 429–442, vol. 12, no. 3, June 2004.
[2] H. Schulzrinne and X. Wang, "Incentive-Compatible Adaptation of Internet Real-Time Multimedia," *IEEE Journal on Selected Areas in Communications*, pp. 417–436, vol. 23, no. 2, February 2005.
[3] J. Aweya, M. Ouellette, and D. Y. Montuno, "Design and stability analysis of a rate control algorithm using the Routh-Hurwitz stability criterion," *IEEE/ACM Transactions on Networking*, pp. 719–732, vol. 12, no. 4, August 2004.
[4] ——, "A simple, scalable and provably stable explicit rate computation scheme for flow control in computer networks," *Int. J. Commun. Syst.*, pp. 593–618, vol. 14, no. 6, August 2001.
[5] D. Bertsekas and R. Gallager, *Data Networks, 2nd Edition*. Prentice-Hall, 1992.
[6] M. Allalouf and 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.
[7] A. T. Campbell and R. R.-F. Liao, "Dynamic Edge Provisioning for Core IP Networks," in *Proc. IEEE/IFIP Int'l Workshop on Quality of Service IWQOS*, Pittsburgh, USA, June 2000.
[8] T. Ahmed, R. Boutaba, and A. Mehaoua, "A Measurement-Based Approach for Dynamic QoS Adaptation in DiffServ Network," *Journal of Computer Communications, Special issue on End-to-End Quality of Service Differentiation, Elsevier Science*, 2004.
[9] E. W. Fulp and D. S. Reeves, "Bandwidth Provisioning

and Pricing for Networks with Multiple Classes of Service," *Computer Networks*, pp. 41–52, vol. 46, no. 1, 16 September 2004.

[10] M. Mahajan, M. Parashar, and A. Ramanathan, "Active Resource Management for the Differentiated Services Environment," *International Journal of Network Management*, pp. 149–165, vol. 14, no. 3, May 2004.

[11] H. T. Tran and T. Ziegler, "Adaptive bandwidth provisioning with explicit respect to QoS requirements," *Computer Communications*, pp. 1862–1876, vol. 28, no. 16, 3 October 2005.

[12] L. Breslau and S. Shenker, "Best-Effort versus Reservations: A Simple Comparative Analysis," in *Proc. ACM SIGCOMM*, September 1998, pp. 3–16.

[13] H. Schulzrinne and X. Wang, "RNAP: A resource negotiation and pricing protocol," in *Int. Workshop Netw. Oper. Syst. Support Digital Audio Video*, Basking Ridge, NJ, June 1999, pp. 77–93.

[14] A. Charny and K. K. Ramakrishnan, "Time Scale Analysis of Explicit Rate Allocation in ATM Networks," in *Proceedings of IEEE INFOCOM*, April 1996.

[15] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, pp. 33–37, vol. 8, 1997.

[16] K. Pawlikowski, H.-D. J. Jeong, and J.-S. R. Lee, "On credibility of Simulation Studies of Telecommunications Networks," *IEEE Communications Magazine*, pp. 132–139, January, 2002.

[17] R. J. La and V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE/ACM Transactions on Networking*, pp. 272–286, vol. 10, no. 2, April 2002.

Jocelyne Elias received her Master of Computer Sciences and Telecommunications Engineering from the Lebanese University of Tripoli in 2002, the Masters Degree (DEA) in Advanced Networks of Knowledge and Organization from University of Technology of Troyes in 2003, and the PhD 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 PhD 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 PhD 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 network, 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). Dr. Pujolle is an Editor for the IEEE Tutorial and Survey.