Algorithms for IP network design with end-to-end QoS constraints

Emilio C.G. Wille a, Marco Mellia b,*, Emilio Leonardi b, Marco Ajmone Marsan b

a Federal Technology Education Center of Paraná, Department of Electronics, Av. Sete de Setembro 3165, Curitiba (PR), Brazil
b Politecnico di Torino, Corso Duca degli Abruzzi, 24-10129 Torino, Italy

Available online 5 October 2005

Abstract

The new generation of packet-switching networks is expected to support a wide range of communication-intensive real-time multimedia applications. A key issue in the area is how to devise reasonable packet-switching network design methodologies that allow the choice of the most adequate set of network resources for the delivery of a given mix of services with the desired level of end-to-end Quality of Service (e2e QoS) and, at the same time, consider the traffic dynamics of today's packet-switching networks. In this paper, we focus on problems that arise when dealing with this subject, namely Buffer Assignment (BA), Capacity Assignment (CA), Flow and Capacity Assignment (FCA), Topology, Flow and Capacity Assignment (TCFA) problems. Our proposed approach maps the end-user's performance constraints into transport-layer performance constraints first, and then into network-layer performance constraints. This mapping is then considered together with a refined TCP/IP traffic modeling technique, that is both simple and capable of producing accurate performance estimates, for general-topology packet-switching design networks subject to realistic traffic patterns. Sub-problems are derived from a general design problem and a collection of heuristic algorithms are introduced for compute approximate solutions. We illustrate examples of network planning/dimensioning considering Virtual Private Networks (VPNs).

© 2005 Elsevier B.V. All rights reserved.

Keywords: Packet-switching networks design and planning; TCP/IP; Queueing theory; Mathematical programming/optimization; Heuristic methods

1. Introduction

The new generation of packet-switching networks is expected to support a wide range of communication-intensive real-time multimedia applications. These applications will have their own different quality-of-service (QoS) requirements in terms of throughput, reliability, and bounds on end-to-end (e2e) delay, jitter, and packet-loss ratio. It is technically a challenging and complicated problem to deliver multimedia information in a timely, synchronized manner over a decentralized, shared network environment, especially one that was originally designed for best-effort traffic such as the Internet.
Accordingly, a key issue in this area is how to devise reasonable packet-switching network design methodologies that allow the choice of the most adequate set of network resources for the delivery of a given mix of services with the desired level of e2e QoS and, at the same time, consider the traffic dynamics of today's packet-switching networks.

The traditional approaches to optimal design and planning of packet networks, extensively investigated in the early days of packet-switching networks [1,2], focus on the network-layer infrastructure thus neglecting e2e QoS issues, and Service Level Agreement (SLA) guarantees. From the end-user's point of view, QoS is driven by e2e performance parameters, such as data throughput, Web page latency, transaction reliability, etc. Matching the user-layer QoS requirements to the network-layer performance parameters is not a straightforward task. The QoS perceived by end-users in their access to Internet services is mainly driven by the Transmission Control Protocol (TCP), the reliable transport protocol of the Internet, whose congestion control algorithms dictate the latency of information transfer. Indeed, it is well known that TCP accounts for a great amount of the total traffic volume in the Internet [3,4], and among all the TCP flows, a vast majority is represented by short-lived flows (also called mice), while the rest is represented by long-lived flows (also called elephants).

The description of traffic patterns inside the Internet is a particularly delicate issue, since it is well known that IP packets do not arrive at router buffers following a Poisson process [5]. Traditionally, either $M/M/1$ or $M/M/1/B$ queueing models were considered as good representations of packet queueing elements in the network. However, the traffic flowing in IP networks is known to exhibit Long Range Dependent (LRD) behavior, which cause queue dynamics to severely deviate from the above model predictions. For these reasons, the usual approach of modeling packet-switching networks as networks of $M/M/1$ queues [6–8] appears now inadequate for the design of such networks. Recently, in [9], the authors for the first time abandon the Markovian assumption in favor of a fractional Brownian motion model, i.e., an LRD traffic model. They solve the discrete Capacity Assignment problem under network e2e delay constraints only, using a simulated annealing meta heuristic. Unfortunately, it is difficult to extend this approach to consider more general network problems, because the relation among traffic, capacity and queueing delay is not expressed as a closed-form expression.

Additionally, with the enormous success of the Internet, all enterprises have become dependent upon networks or networked computation applications. In this context the loss of network services is a serious outage, often resulting in unacceptable delays, loss of revenue, or temporary disruption. To avoid loss of network services, communications networks should be designed so that they remain operational and maintain as high a performance level as feasible, even in the presence of network component failures.

In this paper, we focus on several types of problems that arise when dealing with packet-switching network design. We consider the traffic dynamics of packet networks, as well as the effect of protocols at the different layers of the Internet architecture on the e2e QoS experienced by end-users. Of course, in any realistic network problem an “optimal design” is an extremely difficult task. In [10,11] an IP network design methodology is proposed which is based on a “Divide and Conquer” approach, in the sense that it corresponds to several tasks. Fig. 1 shows the flow diagram of the design methodology. Shaded, rounded boxes represent function blocks, while white parallelograms represent input/output of functions. There are three main blocks, which correspond to the classic blocks in constrained optimization problems: constraints (on the left), inputs (on the bottom right) and optimization procedure (on the top right). Considered as constraints, for every source/destination pair, are the specification of user-layer QoS parameters. Thanks to the definition of the QoS translators, all the user-layer constraints are then mapped into lower-layer constraints, down to the IP layer. The optimization procedure takes as inputs (in accordance to the problem to be solved) the description of the physical topology, the routing algorithm specification, the traffic matrix, and the expression of the cost as function of the design variables. The objective of the optimization is to find the minimum-cost solution that satisfies the user-layer QoS constraints. A second important point of the proposed methodology is the adoption of a refined TCP/IP traffic modeling technique that is both simple and capable of producing accurate performance estimates for packet-switching networks subject to realistic traffic patterns. The main idea behind this approach corresponds to reproducing the effects of traffic correlations on network queueing
elements by means of Markovian queueing models with batch arrivals. Hence, using $M/M/1$ like queues.

The rest of the paper is organized as follows. Section 2 briefly describes the QoS translation problem as well as the traffic and queueing models. Section 3 outlines the general design problem and provides the formulation of the related optimization subproblems. It also introduces heuristic algorithms to compute approximate solutions, and discusses numerical and simulation results. Finally, Section 4 summarizes the main results obtained in this research.

2. QoS translation and models

In this section we describe the QoS translation problem as well as the traffic and queueing models (focusing on the TCP protocol) [10,11].

2.1. QoS translators

The process of translating QoS specifications between different layers of the protocol stack is called QoS translation. According to the Internet protocol architecture, at least two QoS translating procedures should be considered; the first one translates the application-layer QoS constraints into transport-layer QoS constraints, and the second translates transport-layer QoS constraints into network-layer QoS constraints.

2.1.1. Application-layer QoS translator

This module takes as input the application-layer QoS constraints, such as Web page transfer latency, data throughput, audio quality, etc. Given the multitude of Internet applications it is not possible to devise a generic procedure to solve this problem. Hence, in this paper we will focus on ad-hoc solutions depending on the application.

2.1.2. Transport-layer QoS translators

The translation from transport-layer QoS constraints to network-layer QoS parameters, such as Round Trip Time ($RTT$) and packet loss probability ($P_{\text{loss}}$), is more difficult. This is mainly due to the complexity of the TCP protocol, which implements error, flow and congestion control algorithms. The TCP QoS translator accepts as inputs either the maximum file transfer latency ($L_t$), or the minimum file transfer throughput ($T_h$). We require that all flows shorter than a given threshold (i.e., mice) meet the maximum file transfer latency constraint, while longer flows (i.e., elephants) are subjected to the throughput constraint. Obviously, more stringent constraints among latency and throughput will be considered.

The approach is based on the numerical inversion of analytic TCP models, taking as input either the file transfer throughput or latency, and obtaining as outputs $RTT$ and $P_{\text{loss}}$. Among the many models of TCP presented in the literature, we used the TCP latency model described in [12]. We will
refer to this model as the CSA model (from the authors’ names). When considering throughput, we instead exploit the formula in [13], referred as the PFTK model (from the authors’ names). Here, the numerical inversion is just a root finding procedure. There are at least two parameters that affect TCP performance, i.e., $RTT$ and $P_{loss}$. We decided to fix the $P_{loss}$ parameter, and leave $RTT$ as the free variable. This choice is due to the fact that the loss probability has a larger impact on the latency of very short flows, and that it may impact the network load due to retransmissions. Therefore, after choosing a value for $P_{loss}$, a set of curves can be derived, showing the behavior of $RTT$ as a function of file latency and throughput.

As one example, we consider a mixed traffic scenario where data files are exchanged with the file size distribution related in [4]. This distribution, obtained by one-week long measurements, says that 85% of all TCP flows are shorter than 20 packets. Considering this distribution, given the file transfer latency and a fixed throughput of 512 kbps constraint, the curves of Fig. 2 report the maximum admissible $RTT$ which satisfies the most stringent constraint for different values of $P_{loss}$.

### 2.2. Traffic and queueing models

In order to obtain a useful formulation of the optimization problems, it is necessary on one side to be accurate in the prediction of the performance metrics of interest (average delay, packet-loss probability), while on the other side limiting the complexity of the model, (i.e., we adopt models allowing a simple closed-form solution).

The representation of traffic patterns inside the Internet is a particularly delicate issue, since it is well known that IP packets do not arrive at router buffers following a Poisson process [5]. Instead, a high degree of correlation exists, which can be partly due to the TCP control mechanisms. In [14], a simple and quite effective expedient was proposed to accurately predict the performance of network elements subject to TCP traffic, using Markovian queueing models. The main idea behind this approach consists in reproducing the effects of traffic correlations on network queueing elements by means of Markovian queueing models with batch arrivals. The choice of using batch arrivals following a Poisson process has the advantage of combining the nice characteristics of Poisson processes (analytical tractability in the first place) with the possibility of capturing the burstiness of the TCP/IP traffic. Hence, we model network queueing elements using $M[X]/M/1$ like queues. The batch size varies between 1 and $W$ with distribution $[X]$, where $W$ is the maximum TCP window size expressed in segments. The distribution $[X]$ is obtained considering the number of segments that TCP sources send in one $RTT$ for a given file size distribution [14]. The Markovian assumption for the batch arrival process is mainly justified by the Poisson assumption for the TCP connection generation process, as well as the fairly large number of TCP connections simultaneously present in the network. Given the file size distribution, a stochastic model of TCP (described in [14]) is used to obtain the batch size distribution $[X]$. The distribution $[X]$ is obtained only once before starting the optimization process.

### 2.3. Virtual private networks

Designing a packet-switching network today may have quite different meanings, depending on the type of network that is being designed. If we consider the design of the physical topology of the network of a large Internet Service Provider (ISP), the design must very carefully account for the existing infrastructure, for the costs associated with the deployment of a new connection or for the upgrade of an existing link, and for the very coarse granularity in the data rates of high-speed links. Instead, if we consider the design of a corporate Virtual Private Network (VPN), then connections are leased from a long distance carrier, the set of leased lines.

![Fig. 2. RTT constraints as given by the transport-layer QoS translator.](image)
is not a critical component, costs are directly derived from the leasing fees, and the data rate granularity is much finer. While the general methodology for packet-network design and planning, as described in Section 1, can be applied to both contexts, as well as others, in this paper we concentrate on the design of corporate VPNs.

3. Problem statement

The network infrastructure is represented by a graph \( G = (V, E) \) in which \( V \) is a set of nodes (with cardinality \( n \)) and \( E \) is a set of edges (with cardinality \( m \)). A node represents a network router and an edge represents a physical link connecting one router to another. The output interfaces of each router is modeled by a queue with finite buffer. Each network link is characterized by a set of attributes which principally are the flow, the capacity and the buffer size. For a given link \((i,j)\), the flow \( f_{ij} \) is defined as the quantity of information transported by this link, while its capacity \( C_{ij} \) is a measure of the maximal quantity of information that it can transmit. Flow and capacity are expressed in bits per second (bps). Each buffer can accommodate a maximum of \( B_{ij} \) packets, and \( d_{ij} \) is the link physical length.

Considering the \( M_{[X]}/M/1/\infty \) queue, the average packet delay is given by the following expression [15] (where we drop the subscript \((ij)\) for simplicity):

\[
E[T] = \frac{K}{\lambda} \frac{\rho}{1-\rho} = \frac{K}{\mu} \frac{1}{C-f}
\]  
with \( K \) given by
\[
K = \frac{m_{[X]} + m_{[X]}^1}{2m_{[X]}},
\]
where \( \rho = f/C \) is the link utilization factor, the packet length is assumed to be exponentially distributed [1,2] with mean \( 1/\mu \) (bits/packet), \( \lambda = \mu f \) is the arrival rate (packets/s), and \( m_{[X]} \) and \( m_{[X]}^1 \) are the first and second moments of the batch size distribution [X].

The average traffic requirements between nodes are represented by a traffic matrix \( \mathbf{T} = \{T_{ij}\} \), where the traffic \( T_{sd} \) between a node pair \((s,d)\) represents the average number of bps sent from source \( s \) to destination \( d \). We consider as traffic offered to the network \( T_{sd} = \hat{\gamma}_{sd}/(1-P_{\text{loss}}) \), to take into account the retransmissions due to the losses that flows experience along their path to the destination. The flow of each link that composes the topological configuration depends on the traffic matrix. We consider that for each source/destination pair \((s,d)\), the traffic is transmitted over exactly one directed path in the network. The routing and the traffic uniquely determine the vector \( \mathbf{f} = (f_1, f_2, \ldots, f_m) \) where \( f \) is a multi-commodity flow for the traffic matrix; it must obey the law of flow conservation. A multi-commodity flow results from the sum of single-commodity flows \( f^{kl} \), where \( f^{kl} \) is the average flow generated by packets with source node \( k \) and destination node \( l \).

Now we can state the general network design problem as follows: consider that we are given the locations of the network routers, the traffic flow requirements, and the link and buffer costs. Our design task is to choose a topology, to select the capacity of the links in this topology, and to design a routing procedure for the traffic from its origins to its destinations, in a way which optimizes an objective function while meeting all the system (QoS and reliability) constraints. As reliability constraint we consider that all traffic must be exchanged even if a single node fails (2-connectivity), and the QoS constrains correspond to maintaining the e2e packet delay for each network source/destination pair below a maximum tolerable value. When explicitly considering TCP traffic it is also necessary to tackle the Buffer Assignment (BA) problem, which corresponds to dimension buffer sizes subject to packet-loss probability constraints.

The above stated problem is intractable. The number of topologies to consider is too large and, in addition, we have a multi-commodity flow problem. Subproblems can be derived from this general problem and solved separately, in such a way so as to obtain feasible solutions to the general problem. Hence, we may now define three subproblems that differ only in the set of permissible design variables. It is important to note that for a given subproblem a specific optimization technique must be applied to solve it.

3.1. The Capacity Assignment problem

In this subsection we focus on the Capacity Assignment (CA) problem, i.e., the selection of the link capacities. The decision of fixing a priori the loss probability allows us to decouple the CA problem from the BA problem. We first solve the CA problem considering the e2e delay constraints only. Then, we enforce the loss probability to meet the \( P_{\text{loss}} \) constraints by properly choosing buffer sizes.
In the first optimization, a $M_{\infty}/M/1/\infty$ queueing model is used, i.e., a queueing model with infinite buffers. This provides a pessimistic estimate of the queueing delay that packets suffer with finite buffers, which will result from the second optimization step, during which an $M_{\infty}/M/1/B$ queueing model is used.

Different formulations of the CA problem result by selecting (i) the cost functions, (ii) the routing model, and (iii) the capacity constraints. In the VPN case common assumptions are (i) linear costs, (ii) non-bifurcated routing, and (iii) continuous capacities.

Our goal is to determine the link capacities in order to minimize the network cost subject to the maximum allowable e2e packet delay. Given the network topology, the traffic requirements, and the routing, the CA problem is formulated as the following optimization problem:

$$Z_{CA} = \min \sum_{i,j} g(d_{ij}, C_{ij})$$  \hspace{1cm} (3)

subject to:

$$K_{i} \sum_{i,j} \frac{\delta_{ij}^{ad}}{C_{ij} - f_{ij}} \leq RTT_{sd} - \tau_{sd} - \tau_{ds} \ \forall (s,d),$$  \hspace{1cm} (4)

$$f_{ij} = \sum_{s,d} \delta_{ij}^{ad} \ \forall (i,j),$$  \hspace{1cm} (5)

$$C_{ij} \geq f_{ij} \geq 0 \ \forall (i,j).$$  \hspace{1cm} (6)

The objective function (3) represents the total link cost, which is a linear function of both the link capacity and the physical length, i.e., $g(d_{ij}, C_{ij}) = d_{ij} C_{ij}$. Eq. (4) is the e2e packet delay constraint for each source/destination pair. It says that the total amount of delay experienced by all the flows routed on a path should not exceed the maximum RTT (see Section 2.1) minus the propagation delay of the route. $\delta_{ij}^{ad}$ is an indicator function which is one if link $(i,j)$ is in path $(s,d)$ and zero otherwise. Non-bifurcated routing model is used where the traffic will follow exactly one path from source to destination. Eq. (5) defines the average data flow on the link. Constraints (6) are non-negativity constraints. Finally, $K_{i} = K/\mu$.

We notice that the above stated CA problem is a convex optimization problem, and its global minimum can be found using standard convex programming techniques, for example, the logarithm barrier method [16]. However, these algorithms are time-consuming. A fast suboptimal solution to this problem can be found using the following heuristic.

### 3.1.1. Suboptimal solution to the CA problem

A simple heuristic can be derived to obtain solutions to the CA problem. The main idea is to decompose the problem into $n \times (n - 1)$ single constrained problems (one for each path $(s,d)$). Let $I_{sd}$ be the set of links which compose path $(s,d)$, and let $C_{ij}^{ad}$ be an auxiliary variable which corresponds to the capacity of the link $(i,j)$ when considering the path $(s,d)$. To solve each single path problem we apply the Lagrangean multiplier method obtaining:

$$L(\psi) = \min \left[ \sum_{(i,j) \in I_{sd}} d_{ij} C_{ij}^{ad} + \psi \left( \sum_{(i,j) \in I_{sd}} \frac{1}{C_{ij}^{ad} - f_{ij} - b_{sd}} \right) \right],$$  \hspace{1cm} (7)

subject to:

$$C_{ij}^{ad} \geq f_{ij} \geq 0 \ \forall (i,j), \forall (s,d),$$  \hspace{1cm} (8)

where

$$b_{sd} = \frac{1}{K_{1}} (RTT_{sd} - \tau_{sd} - \tau_{ds}) \ \forall (s,d).$$  \hspace{1cm} (9)

The solutions to this problem are given by

$$C_{ij}^{ad} = f_{ij} + \sum_{(k,l) \in I_{sd}} \sqrt{d_{kl}} \frac{b_{sd}}{b_{sd} \sqrt{d_{ij}}}.\hspace{1cm} (10)$$

Knowing the values for the variables $C_{ij}^{ad}$ (in the single path problem) we obtain admissible values for the capacities $C_{ij}$ (in the original CA problem) assigning:

$$C_{ij} = \max_{s,d} \{C_{ij}^{ad}\}.\hspace{1cm} (11)$$

### 3.2. The Buffer Assignment problem

A second step corresponds to dimension buffer sizes, i.e., to solve the following optimization problem:

$$Z_{BA} = \min \sum_{i,j} h(B_{ij})$$  \hspace{1cm} (12)

Subject to:

$$\sum_{j} \delta_{ij}^{ad} p_{ij} (B_{ij}, C_{ij}, f_{ij}, \{X\}) \leq P_{loss}, \ \forall (s,d),$$  \hspace{1cm} (13)

$$B_{ij} \geq 0, \ \forall (i,j).$$  \hspace{1cm} (14)

The objective function (12) represents the total buffer cost, which is the sum of the buffer cost functions, $h(B_{ij}) = B_{ij}$. Eq. (13) is the loss probability constraint for each source/destination node pair. It says that the total loss probability experienced by
all the flows routed on the path \((s,d)\) should not exceed the maximum fixed \(P_{\text{loss}}\). Here, \(p(B_{ij}, C_{ij}, f_{ij}, [X])\) is the average loss probability for the \(M_{\{X\}}/M/1/B\) queue, which is evaluated by solving its Continuous Time Markov Chain (CTMC). Constraints (14) are non-negativity constraints.

In the previous formulation we have considered the following upper bound on the value of loss probability for path \((s,d)\) (constraint (13)).

\[
P_{\text{loss}} = 1 - \prod_{i,j} \left( 1 - \delta_{ij}^d p(B_{ij}, C_{ij}, f_{ij}, [X]) \right) \leq \sum_{i,j} \delta_{ij}^d p(B_{ij}, C_{ij}, f_{ij}, [X]).
\] (15)

Notice also that the first part of Eq. (15) is based on the assumption that link losses are independent. Consequently, the solution of the BA problem is a conservative solution to the full problem.

The above stated BA problem is a convex optimization problem [10], and its global minimum can be found using standard convex programming techniques [16].

### 3.2.1. Numerical examples and simulations

In this section we present some numerical results, which correspond to the solution of selected CA and BA problems (here we used the logarithm barrier method). In order to validate our designs, we ran simulation experiments using the ns-2 [17], software package.

As a first example, we present results obtained considering the multi-bottleneck mesh network shown in Fig. 3. The network topology comprises 5 nodes and 12 links. In this case, link propagation delays are all equal to 0.5 ms, that correspond to a link length of 150 km. Fig. 3 indicates link identifiers, link routing weights (in parentheses), and traffic requirements. Routing weights are chosen in order to have one single path for every source/destination pair.

We consider a mixed traffic scenario where the file size (ranging from 1 to 195 packets) follows the distribution related in [4]. We choose, for this case, the following TCP QoS constraints: (i) latency \(L_t \leq 0.5\) s for files shorter than 20 packets, (ii) throughput \(T_h \geq 512\) kbps for files longer than 20 packets, and (iii) \(P_{\text{loss}} = 0.01\), using the transport-layer QoS translator we obtain the equivalent network-layer performance constraint \(R_T \leq 0.07\) s (for the sake of simplicity, in the examples we will consider \(R_{T,s} = R_T, \forall (s,d)\)).

The CA and BA problems associated with this network have 12 unknown variables and 11 constraint functions (we have discarded nine redundant constraint functions). In order to obtain some comparisons, we also implemented a design procedure using the classical formula (see [1,2]) which considers an \(M/M/1\) queue model in the CA problem. We also extended the classical approach to the BA problem, which is solved considering \(M/M/1/B\) queues. We also imposed these same constraints in the classical approach. In Fig. 4, it can be immediately noticed that considering the burstiness of IP traffic radically changes the network design. The link utilization factors have an average equal to about \(\bar{\rho} = 0.8\), and buffer sizes have average \(\bar{B} = 175\), which is about 4 times the average number of packets in the queue (40 packets). Indeed, the link utilizations obtained with our methodology are much smaller than those produced by the classical approach, and buffers are much larger. This is due to the bursty arrival process of IP traffic, which is well captured by the \(M_{\{X\}}/M/1/B\) model.

---

**Fig. 3. 5-Node network: topology and traffic requirements.**
To validate the design methodology, we ran ns-2 simulations for drop-tail and RED buffers. We assume that New Reno is the TCP version of interest. In addition, we assume that TCP connections are established choosing at random a server-client pair, and are opened at instants described by a Poisson process. Connection opening rates are determined so as to set the link flows to their desired values. The packet size is assumed constant, equal to the maximum segment size (MSS); the maximum window size is assumed to be 32 segments.

We report detailed results selecting traffic from node 4 to node 1, which is routed over one of the most congested paths (three hops, over links: 8, 7, 6). Fig. 5 plots the file transfer latency for all file sizes for the selected source/destination pair. The QoS constraint of 0.5 s for the maximum latency is also reported. We can see that model results and simulation estimates are in perfect agreement with specifications, the constraints being perfectly satisfied for all files shorter than 20 packets. Note also that longer flows obtain a much higher throughput than the target, because the file transfer latency constraint is more stringent (as also shown in Fig. 2). It is important to observe that a network dimensioned using the classical approach cannot satisfy all the QoS constraints.

As a second example of multi-bottleneck topology we chose a network comprising 10 nodes and 24 links. Link propagation delays are uniformly distributed between 0.05 and 0.5 ms, i.e., link lengths vary between 15 km and 150 km. The traffic requirement matrix is set to obtain an average link flow of about 15 Mbps. The CA and BA problems associated with this network have 24 unknown variables and 66 constraint functions (we have discarded 24 redundant constraint functions). We considered the same design target QoS parameters as for the previous example. In order to observe the impact of traffic load and performance constraints on our design methodology, we performed several numerical experiments.

Fig. 6 shows the range of network link utilization as a function of traffic load (first plot). Looking at how traffic requirements impact the CA problem, we observe that the larger the traffic load, the higher the utilization factor. This is quite intuitively explained by a higher statistical multiplexing gain, and by the fact that the RTT is less affected by the transmission delay of packets at higher speed. The behavior of buffer sizes as a function of traffic requirements is shown on the second plot. As expected, the larger the traffic load, the higher the space needed in queue (buffer sizes).

\footnote{Optimal values for RED parameters are obtained according to the procedure given in \cite{10,11}.}
The impact of more stringent QoS requirements is considered in Fig. 7 ($P_{\text{loss}} = 0.01$, average link traffic = 15 Mbps). Notice that, in order to satisfy a very tight constraint (file latency $L_t \leq 0.2$ s), it is necessary to have an utilization factor close to 20% on some particularly congested links (first plot). Tight constraints mean packet delays with small values and thus larger capacity values concerning the link flows. On the contrary, relaxing the QoS constraints, we note a general increase in the link utilization, up to 90%. The behavior of buffer sizes as a function of file transfer latency requirements is shown in the second plot. We see that stringent QoS requirements force small values for buffer sizes.

Finally, Fig. 8 shows link utilization and buffer sizes considering different packet-loss probability constraints, while keeping fixed the file transfer latency $L_t \leq 2$ s and throughput $T_h \geq 512$ kbps (average link traffic = 15 Mbps). Obviously, an increase of $P_{\text{loss}}$ values forces the transport-layer QoS translator to reduce the $RTT$ to meet the QoS constraints. As a consequence, the utilization factor decreases (first plot).

More interesting is the effect of selecting different values of $P_{\text{loss}}$ on buffer sizes (second plot). Indeed, to obtain $P_{\text{loss}} \leq 0.005$, buffer sizes longer than 350 packets are required, while $P_{\text{loss}} \leq 0.02$ can be guaranteed with buffers shorter than 70 packets. This result stems from the correlation of TCP traffic and is not captured by a Poisson model.

Simulations using ns-2 confirm that the target QoS parameters are met in all cases.

### 3.3. The capacity and flow assignment problem

Traditionally, packet-network design focused on optimizing either network cost or performance by tuning link capacities and routing strategies. Since the routing and link capacities optimization problems are closely interrelated, it is appropriate to jointly solve them in what is called the Capacity and Flow Assignment (CFA) problem.
Our goal is to determine a route for the traffic that flows on each source/destination pair and the link capacities in order to minimize the network cost subject to the maximum allowable e2e packet delay. Let \( j_{sd} \) be a decision variable which is one if link (\( i, j \)) is in path (\( s, d \)) and zero otherwise. Thus the CFA problem is formulated as the following optimization problem:

\[
Z_{CFA} = \min \sum_{i,j} g(d_{ij}, C_{ij})
\]

subject to:

\[
\sum_{j} \kappa_{ij}^{sd} - \sum_{j} \kappa_{ji}^{sd} = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = t \\
0 & \text{otherwise} 
\end{cases} \quad \forall(i,s,d) \tag{17}
\]

\[
K_1 \sum_{i,j} \frac{\kappa_{ij}^{sd}}{C_{ij} - f_{ij}} \leq RTT_{sd} - K_2 \sum_{i,j} \kappa_{ij}^{sd}d_{ij} \quad \forall(s,d) \tag{18}
\]

\[
f_{ij} = \sum_{s,d} \kappa_{ij}^{sd} \quad \forall(i,j) \tag{19}
\]

The objective function (16) represents total link cost, which is a linear function of both the link capacity and the physical length. Constraint set (17) enforces flow conservation, defining a route for the traffic from a source \( s \) to a destination \( d \). Eq. (18) is the e2e packet delay constraint for each source/destination pair. Eq. (19) defines the average data flow on the link. Constraints (20) and (21) are non-negativity and integrality constraints, respectively. Finally, \( K_1 = K/\mu \) and \( K_2 \) is a constant to convert distance into time.

We notice that this problem is a nonlinear non-convex mixed-integer programming problem. Other than the nonlinear constraint (18), it is basically a multi-commodity flow problem [19]. Multi-commodity flow belongs to the class of NP-hard problems for which no known polynomial-time algorithms exist [20]. In addition, thanks to its non-convex property there are in general a large number of local minima solutions. Therefore, in this paper we only discuss CFA suboptimal solutions.

In [11] we proposed a composite upper and lower bounding procedure based on a Lagrangean relaxation [22] of the CFA problem. The purpose is to obtain a relaxed problem, called Lagrangean subproblem, which is easier to solve than the original problem. The objective value from the Lagrangean relaxation problem provides a lower bound (LB), in the case of minimization, for the optimal solution to the original problem. The best lower bound can be derived by solving the Lagrangean dual. To solve the dual problem we used a subgradient optimization technique [23]. Information obtained from the Lagrangean relaxation is then used by application-dependent heuristics to construct feasible solutions to the original problem, i.e., a primal heuristic (PH).

In order to permit some comparisons, we also apply a logarithmic barrier CA solution with minimum-hop routing (MinHop + CA), i.e., we just ignore the routing optimization when solving the CA problem. Another approach is described in the following subsection.

### 3.4. The Greedy Weight Flow Deviation method

The classical Flow Deviation (FD) method is well known to solve CFA problems [1,2]. In this section
we present a heuristic, based on the FD method, to solve the CFA problem presented in Section 3.3.

Considering that no closed-form expression for the optimal capacities can be derived from our CFA formulation we proceeded in the following way: first, it is straightforward to show that the link weights in the original FD method are given by

\[ L_{ij} = \frac{d_{ij} C_{ij}}{f_{ij}} \] (22)

Second, in order to enforce e2e QoS delay performance constraints, the link capacities \( C_{ij} \) must be obtained using the CA solver presented in Section 3.1.1. As our new method relies on the greedy nature of the CA solver algorithm to direct computations toward a local optima, we called it the Greedy Weight Flow Deviation (GWFD) method.

As noted before the CFA problem admits several local minima. A way to obtain a more accurate estimate of the global minima is to restart the procedure using random initial flows. However, we obtained very good results setting as initial trail \( L_{ij} = d_{ij} \).

The following is a description in pseudo-code of the GWFD method:

**Greedy Weight Flow Deviation method:**

Given: feasible \( f^0 \) and \( C^0 \); \( f^* = f^0 \); \( C^* = C^0 \); \( p = 0 \)

Repeat

1. Compute link weights \( L^p \)
2. Compute minimum-weight paths
3. Compute flows \( f^{p+1} \)
4. Solve CA problem and obtain \( C^{p+1} \)
5. If \( D(C^{p+1}) \geq D(C^p) \) Stop
   Else
     (a) \( f^* = f^{p+1} \); \( C^* = C^{p+1} \)
     (b) \( p = p + 1 \)
   End Else
End Repeat

It must be noted that the problem represented by the formulation (16)–(21) and the problem addressed by the GWFD algorithm are not exactly the same. In fact, the traffic routing solutions resulting from the GWFD algorithm are minimum-weight paths, and those resulting from the CFA formulation are not necessarily minimum-weight paths.

### 3.4.1. Numerical examples

In this section we present results obtained considering several fixed topologies (40-node, 160-link each), which have been generated using the BRITE topology generator [21] with the router level option. Link propagation delays are uniformly distributed between 0.5 ms and 1.5 ms, i.e., link lengths vary between 100 km and 300 km. Random traffic matrices were generated by picking the traffic intensity of each source/destination pair from a uniform distribution. The average source/destination traffic requirement was set to \( \gamma_{sd} = 5 \text{ Mbps} \). For all source/destination pairs, the target QoS constraints are: (i) latency \( L_t \leq 0.2 \text{ s} \) for files shorter than 20 segments, (ii) throughput \( T_{th} \geq 512 \text{ kbps} \) for files longer than 20 segments, and (iii) \( P_{loss} = 0.001 \).

Using the transport-layer QoS translator (Section 2), we obtain the equivalent network-layer performance constraint \( RTT \leq 0.032 \text{ s} \) for all source/destination node pairs.

Our goal is to obtain routing and link capacities. For each topology, we solved both the related CFA and BA problems. Fig. 9 shows network costs for 10 different topologies. The GWFD solutions are compared to solutions from other three techniques (LB, PH, and MinHop + CA) [11].

We can observe that the GWFD solutions, for all considered topologies, always fall rather close to the lower bound (LB). The gap between GWFD and LB is about 13%. In addition, the GWFD algorithm is faster than the primal heuristic approach (PH)—only 5 s of CPU time are needed to solve an instance with 40 nodes—while it obtains very similar results.

Avoiding to optimize the flow assignment sub-problem results in more expensive solutions, as
shown by the “Min Hop” routing associated with an optimized CA problem. This underlines the need to solve the CFA problem rather than a simpler CA problem.

A second set of experiments was performed to investigate the impact of the latency constraints on the optimized network cost. Fig. 10 shows the LB and GWFD results for latency constraint values ranging from 0.2 s to 1.0 s. The plots clearly show the trade off between cost and latency; as expected, costs grow when the latency constraints become tighter. It is interesting to observe that when the latency constraints become very tight (latencies become close to zero), the sensitivity of the network cost increases.

3.5. The Topology, Capacity and Flow Assignment problem

In this section the objective is to determine a less expensive solution to interconnect nodes, and assign flow and capacities, while satisfying the reliability and e2e QoS constraints. This problem is called the Topological, Capacity and Flow Assignment (TCFA) problem. This is a complex combinatorial optimization problem, which can be classified as NP-complete [20]. Polynomial algorithms which can find the optimal solution for this problem are not known. Therefore, heuristic algorithms are applied, searching for solutions. We analyze two meta heuristic approaches: the Genetic Algorithm (GA) [24,25], and the Tabu Search (TS) algorithm [26] to address the topological design problem. GAs are heuristic search procedures which apply natural genetic ideas such as natural selection, mutations and survival of the fittest. The TS algorithm is derived from the classical Steepest Descent method, however thanks to an internal mechanism that accepts worse solutions than the best solution so far, it is less subject to local optima entrapment. Details about the application of GA and TS algorithms to topological design can be found in Appendix A.

The TCFA problem can be formulated as follows: given the geographical location of the network nodes on the territory, the traffic matrix, and the capacity costs; minimize the total link cost, by choosing the network topology and selecting link flows and capacities, subject to QoS and reliability constraints. As reliability constraint we consider that all traffic must be exchanged even if a single node fails (2-connectivity). There is a trade off between reliability and network cost; we note that more links between nodes imply more routes between each node pair, and consequently the network is more reliable; on the other hand, the network is more expensive. Finally, the QoS constraints correspond to maintaining the e2e packet delay for each network source/destination pair below a maximum tolerable value.

Our solution approach is based on the exploration of the solution space (i.e., 2-connected topologies) using the meta heuristic algorithms. As the goal is to design a network that remains connected despite one node failure, for each topology evaluation, actually, we construct $n$ different topologies that are obtained from the topology under evaluation by the failure of a node each time, and then for each topology we solve its related CFA problem (using the GWFD method). Link capacities are set to the maximum capacity value found so far considering the set of topologies. Using the capacities obtained, the objective function (network cost) is obtained.

3.5.1. Numerical examples and simulations

In this section we present some selected numerical results considering network designs obtained with the meta heuristic approaches. We consider the same mixed traffic scenario where the file size follows the distribution shown in [4].

A first set of experiments was performed to investigate the performance of GA and TS algorithms. As a first example, we applied the proposed methodology to set up a 15-node VPN network over a given 35-node, 140-link physical topology. The target QoS...
constraints for all source/destination pairs are: (i) file latency $L_t \leq 1\ s$ for files shorter than 20 segments, (ii) throughput $T_h \geq 512\ \text{kbps}$ for files longer than 20 segments. Selecting $P_{\text{loss}} = 0.01$, we obtain a network-level design constraint equal to $\text{RTT} \leq 0.15\ s$ for all source–destination pairs. The average source/destination traffic requirement was set to $\bar{c}_{sd} = 3\ \text{Mbps}$. Link lengths vary between 15 km and 300 km (average = 225 km).

Fig. 11 shows the network cost as a function of computational time (in seconds) considering both the GA and TS algorithms (for different values of population ($N_I$) and Tabu list ($T_L$) sizes). We notice that, after a period of 1.38 h, the solution values differ by at most 2.5%. The best solution is given by the GA with $N_I = 160$ individuals (in contrast, the same solution value was reached by the TS, with $T_L = 20$ moves, after a period of 12 h). The GA with a small population quickly stagnates, and its solution value is “relatively poor”. Using a population size of $N_I = 160$ (value suggested by the estimate given in Appendix A.1) the GA makes slower progress but consistently discover better solutions. If the population size increases further, the GA becomes slower, without however, being able to improve the solutions (not shown in the figure).

In the second set of experiments, we analyze the impact of the traffic scenario on the obtained network topology. We consider the dimensioning of VPN network over a given 10-node, 40-link physical topology. The target QoS constraints for all source/destination pairs are the same used in the first experiment; and the link lengths vary between 140 km and 760 km (average = 380 km). Two traffic scenarios are considered.

In the first scenario, source/destination traffic is randomly generated from a uniform distribution with average value $\bar{c}_{sd} = 1\ \text{Mbps}$. Using the GA approach, the final topology is shown in Fig. 12(A). Solid lines correspond to the chosen links that synthesize the VPN network topology (dashed lines are existing links, but they are not chosen for the VPN topology). We notice that several connections are needed to guarantee the network 2-connectivity.

In the second case, traffic relations are set as follows: two nodes offer an average aggregated traffic equal to 5 Mbps (nodes 3 and 6 in Fig. 12), one node offers 2 Mbps (node 4), and the rest offer traffic equal to 1 Mbps. From Fig. 12(B) we see that three new links were added in order to drain off the increased traffic from nodes 3 and 6; while a link between nodes 2 and 5 was removed.

In order to validate the network design, we compare the target performance parameters against the performance measured from very detailed simulation experiments (using the ns-2 simulator). We performed packet-level simulations to check whether the e2e QoS constraints are actually met. In this case, we completed the design of the network shown Fig. 12(A) by solving its associated BA problem with drop-tail buffers.

As one example, we did path simulations considering the path that connects nodes 10, 6, 2, 8 and 3. Table 1 reports the optimal values for capacities and buffer sizes; it also shows link flows values for two working scenarios: (i) normal network operation, and (ii) failure of network node 5 (in this case some links must transport an increased traffic flow, $f'$). We notice that, in this example, the path from nodes 10 to 3 is the same for node 5 working/failure case.
Fig. 13 plots the file transfer latency for all flow size classes for the selected source/destination pair. The QoS constraint of 1 s for the maximum latency is reported, as well as model results. We can see that the results are in perfect agreement with the specifications, the constraints being perfectly satisfied for all files shorter than 20 packets, for both working scenarios. Note also that in the case of normal network operation the file transfer latency has smaller values, resulting from the greater gap between link capacities and flows. As noted before, longer flows obtain a much higher throughput than the desired one.

4. Conclusion

In this paper, we have considered the QoS and reliability design of packet networks, presenting mathematical formulations and introducing a collection of heuristic algorithms to compute approximate solutions. Two important elements are considered in our approach: (a) the mapping of the e2e QoS constraints into transport-layer performance constraints first, and then into network-layer performance constraints; and (b) a refined TCP/IP traffic modeling technique that is both simple and capable of producing accurate performance estimates for general-topology packet-switching networks loaded by realistic traffic patterns. By explicitly considering TCP traffic, we also need to consider the impact of finite buffers, therefore facing the Buffer Assignment problem. To the best of our knowledge, no previous work solves packet-network design problems accounting for user layer e2e QoS constraints considering more realistic traffic models.

The numerical results have shown that the burstiness of IP traffic radically changes the network design. Indeed, the link utilization obtained with our approach is much smaller than those produced by the classical approach, and the buffer values are much longer. This is due to the bursty arrival process of IP traffic, which is well captured by the $M/M/c/0$ model. On the other hand, the capacity assignment using the classical approach cannot satisfy all the QoS constraints.

In addition, network costs can be reduced by the jointly optimization of routing and link capacities since these are closely interrelated. For this scope, we have proposed a new CFA algorithm, called GWFD, that is capable of assigning flow and capacities under e2e QoS constraints. The proposed GWFD method is particularly interesting. It can solve CFA instances in a fast and quite accurate way.

Based on the GWFD method we have proposed a practical, useful way to solve the topological design problem with e2e QoS and reliability constraints. This approach, which considers GA and TS meta heuristics, while not necessarily an original idea, represents a pragmatic solution to the problem. Computational results suggest a better efficiency of the GA approach in providing good solutions for medium-sized computer networks, in comparison with well-tried conventional TS methods.

In order to validate the proposed methodology, we have compared results against detailed simulation experiments (using ns-2 software) in terms of network performance. The target QoS performance is met in all cases.

Acknowledgement

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.
Appendix A. Applying meta heuristic algorithms to network design

A.1. Genetic algorithms

Genetic algorithms (GAs), as powerful and broadly applicable stochastic search and optimization techniques, are the most widely known types of evolutionary computation methods today. The basic principles of GAs were first established rigorously by Holland [24]. Selection, Genetic Operation and Replacement, directly derived by from natural evolution mechanisms are applied to a population of solutions, thus favoring the birth and survival of the best solutions. They are generally good at finding “acceptably good” solutions to problems in “reasonable” computing times.

In general, a genetic algorithm has five basic components as follows: (a) an encoding method, that is a genetic representation (genotype) of solutions to the problem; (b) a way to create an initial population of individuals (chromosomes); (c) an evaluation function, rating solutions in terms of their fitness, and a selection mechanism; (d) the genetic operators (crossover and mutation) that alter the genetic composition of offspring during reproduction; and (e) values for the parameters of the genetic algorithm.

In GAs, a population of \( N \) solutions (individuals) is created initially. Then by using genetic operators a new generation is evolved. The fitness of each individual determines whether it will survive or not. The individuals in the current population are then replaced by their offspring, based on a certain replacement strategy. After a number of generations \( N_G \), or some other criterion is met, it is hoped that a near optimal solution is found.

Genotype: The genetic encoding of an individual is called a genotype and the corresponding physical appearance of an individual is called a phenotype. As we know, a gene in a chromosome is characterized by two factors: locus, the position of the gene within the structure of chromosome, and allele, the value the gene takes. The allele can be encoded as binary, a real number, or other forms and its range is usually defined by the problem.

Manipulation of topologies with the genetic algorithm requires that they are represented in some suitable format. Although more compact alternatives are possible, the connectivity matrix of a graph is an adequate representation. A graph is represented by an \( n \times n \) binary matrix, where \( n \) is the number of nodes of the graph. The value of each element in row \( i \) and column \( j \), a “1” or a “0”, tells whether or not a specific edge connects the pair of nodes \((i,j)\).

Fitness evaluation: The evaluation of the objective function is usually the most demanding part of a GA algorithm because it has to be done for every individual in every generation. In this paper, a fitness function, which is an estimation of the goodness of the solution for the topological design problem, is inversely proportional to the objective function value (cost). The lower the cost, the better the solution is.

Selection: Parent selection emulates the survival-of-the-fittest mechanism in nature. It is important to ensure some diversity among the population by breeding from a selection of the fitter individuals, rather than just from the fittest.

The selection process used here is based on the tournament method. In this approach pairs of individuals are picked at random and the one with the better fitness (the one which “wins the tournament”) is used as one parent. The tournament selection is then repeated on a second pair of individuals to find the other parent from which to breed.

Crossover: Crossover is a recombination operator that combines subparts of two parent chromosomes to produce offspring that contain some parts of both parents’ genetic material. The probability, \( p_c \), that these chromosomes are recombined (mated) is a user-controlled option and is usually set to a high value (e.g., 0.95).

Unfortunately, according to the genotype representation used here, the above crossover operators are not suitable for recombination of two individuals (the crossover operation mostly leads to illegal individuals).

In this paper, we first use a simple one-point crossover operator. Then, an effective and fast check and recovery algorithm is used to repair the illegal individuals. If the repair operation is unsuccessful the parents are considered as crossover outputs.

Mutation: Mutation is used to change, with a fixed small probability, \( p_m \), the value of a gene (a bit) in order to avoid the convergence of the solutions to “bad” local optima. As a population evolves, there is a tendency for genes to become dominant.

Mutation is therefore important to “loosen up” genes which would otherwise become fixed. In our experiments good results were obtained using a mutation operator that simply changes one bit, picked at random, for each produced offspring.
Replacement strategies: In this paper we use the generational-replacement method with elitist strategy where once the sons' population has been generated, it is merged with the parents' population according to the following rule: only the best individuals present in both sons' population and parents' population enter the new population. The elitist strategy may increase the speed of domination of a population by a super chromosome, but on balance it appears to improve the performance.

Population size: GAs work surprisingly well with quite small populations. Nevertheless, if the population is too small, there is an increased risk of convergence to a local optima; the variety that drives a genetic algorithm's progress cannot be maintained. As population size increases, the GA discovers better solutions more slowly; it becomes more difficult for the GA to propagate good combinations of genes through the population and join them together.

A simple estimate of appropriate population size for topological optimization is given by the following expression:

$$N_I \geq \frac{\log(1 - p^i)}{\log \left(1 - \frac{q}{n(n-1)}\right)}$$  \hspace{1cm} (A.1)

where $N_I$ is the population size, $n$ is the number of nodes, $q$ is the number of links, and $p$ the is probability that the optimal links occurs at least once in the population.

The choice of an appropriate value for $q$ is based on the behavior of the topological optimization process, which can change the number of links in order to reduce the network cost. We, therefore, set $q$ as the minimum number of links that potentially can maintain the network 2-connectivity. Some population size values are given in Table A.1.

### A.2. Tabu search algorithm

The second heuristic considered relies on the application of the Tabu Search (TS) methodology [26]. The TS algorithm can be seen as an evolution of the classical local-search algorithm called Steepest Descent, however, thanks to the interior mechanism that also accepts worse solutions than the current one, it is not subject to local minima entrapment.

For each admissible solution, a class of neighbor solutions is defined. A neighbor solution, is defined as a solution that can be obtained from the current solution by applying an appropriate transformation called move. The set of all the admissible moves uniquely defines the neighborhood of each solution. The dimension of the neighborhood is $N_N$. In this paper a simple move is considered: it either removes an existing link or adds a new link between network nodes.

At each iteration of the TS algorithm, all solutions in the neighborhood of the current one are evaluated, and the best is selected as the new current solution. A special rule, the tabu list, is introduced in order to prevent the algorithm from deterministically cycling among already visited solutions. In the tabu list are the last accepted moves and while a move is stored in the it, it cannot be used to generate a new move for a duration of a certain number of iterations. Therefore it may happen that TS continuing the search will select an inferior solution because better solutions are tabu. If a TS move yields a better solution than any encountered so far, its tabu classification can be overridden (aspiration criterion).

The choice of the tabu list size is very important in the optimization procedure: a small size could cause the cyclic visitation of the same solutions while a big one could block the optimization process for many iterations, avoiding a good visit of the solution space.

### References


Emilio C.G. Wille received his degree in Electronic Engineering in February 1989, and a M.Sc. in Electronic and Telecommunications Engineering in July 1991, both from Centro Federal de Educação Tecnológica do Paraná—CEFET/PR (Curitiba-Brazil). Since October 1991, he is with Electronics Department of CEFET/PR as an Assistant Professor. His teaching duties at CEFET/PR compromise graduate and undergraduate-level courses on electronic and telecommunication theory.

From February 2001, until February 2004, he was with the Electronics Department of Politecnico di Torino (Italy), as a Ph.D. student. He was supported by a CAPES Foundation scholarship from the Ministry of Education of Brazil. His research interests are centered upon the application of optimization algorithms for telecommunication networks design and planning, Markov processes, queueing models, and performance analysis of telecommunication systems.

Marco Mellia was born in Torino, Italy, in 1971. He received his degree in Electronic Engineering in 1997, and a Ph.D. in Telecommunications Engineering in 2001, both from Politecnico di Torino. From March to October 1999 he was with the CS department at Carnegie Mellon University as visiting scholar. Since April 2001, he is with Electronics Department of Politecnico di Torino as Assistant Professor. He has co-authored over 50 papers published in international journals and presented in leading international conferences, all of them in the area of telecommunication networks.

He participated in the program committees of several conferences including IEEE Globecom and IEEE ICC.

His research interests are in the fields of All-Optical Networks, Traffic measurement and modeling, QoS Routing algorithms.

Emilio Leonardi was born in Cosenza, Italy, in 1967. He received a Dr. Ing degree in Electronics Engineering in 1991 and a Ph.D. in Telecommunications Engineering in 1995 both from Politecnico di Torino.

He is currently an Assistant Professor at the Dipartimento di Elettronica of Politecnico di Torino.

In 1995, he visited the Computer Science Department of the University of California, Los Angeles (UCLA), in summer 1999 he joined the High Speed Networks Research Group, at Bell Laboratories/Lucent Technologies, Holmdel (NJ); in summer 2001, the Electrical Engineering Department of the Stanford University and finally in summer 2003, the IP Group at Sprint, Advanced Technologies Laboratories, Burlington CA.

He participated in several national and European projects such as MIUR-MQOS, MIUR-PLANET IP, MIUR-IPPO, MIUR-TANGO, IST-SONATA and IST-DAVID. He is also involved in
several consulting and research projects with private industries, including Lucent Technologies-Bell Labs, British Telecom, Alcatel, and TILAB.

He has co-authored over 100 papers published in international journals and presented in leading international conferences, all of them in the area of telecommunication networks. He received the IEEE TCGN best paper award for a paper presented at the IEEE Globecom 2002, “High Speed Networks Symposium”.

He participated in the program committee of several conferences including: IEEE Infocom, IEEE Globecom, and IEEE ICC. He was guest editor of two special issues of IEEE Journal of Selected Areas of Communications focused on high-speed switches and routers.

His research interests are in the field of: performance evaluation of communication networks, all-optical networks, queueing theory, packet switching.

Marco Ajmone Marsan is a Full Professor at the Electronics Department of Politecnico di Torino, in Italy, and the Director of the Institute for Electronics, Information and Telecommunications Engineering of the National Research Council.

He has degrees in Electronic Engineering from Politecnico di Torino and University of California, Los Angeles.

He was at Politecnico di Torino Electronics Department from November 1975 to October 1987—first as a researcher, then as an Associate Professor. He was a Full Professor at the University of Milan Computer Science Department from November 1987 to October 1990. During the summers of 1980 and 1981, he was with the Research in Distributed Processing Group, Computer Science Department, UCLA. During the summer of 1998 he was an Erskine Fellow at the University of Canterbury in New Zealand Computer Science Department.

He has co-authored over 300 journal and conference papers in Communications and Computer Science, as well as the two books “Performance Models of Multiprocessor Systems,” published by the MIT Press, and “Modeling with Generalized Stochastic Petri Nets,” published by John Wiley.

In 1982, he received the best paper award at the Third International Conference on Distributed Computing Systems in Miami, Florida. In 2002, he was awarded a “Honoris Causa” Degree in Telecommunications Networks from the Budapest University of Technology and Economics.

He is a corresponding member of the Academy of Sciences of Torino. He participates in a number of editorial boards of international journals, including the IEEE/ACM Transactions on Networking, and the Computer Networks Journal by Elsevier.

He has been the principal investigator in national and international research projects dealing with telecommunication networks. His current interests are in the performance evaluation of communication networks and their protocols.