Using Partial Differential Equations to Model TCP Mice and Elephants in Large IP Networks

Marco Ajmone Marsan, *Fellow, IEEE*, Michele Garetto, *Member, IEEE*, Paolo Giaccone, *Member, IEEE*, Emilio Leonardi, *Member, IEEE*, Enrico Schiattarella, *Student Member, IEEE*, and Alessandro Tarello, *Student Member, IEEE*

***Abstract*—In this paper we propose a new fluid model approach in which a different description of the dynamics of traffic sources is adopted, exploiting *partial* differential equations. This new description of the source dynamics allows the natural representation of short-lived as well as long-lived TCP connections, with no sacrifice in the scalability of the model. In addition, the use of partial differential equations permits the description of distributions, instead of averages, thus providing better accuracy in the results.**

The comparison between the performance estimates obtained with fluid models and with *ns-2* simulations proves the accuracy of the proposed modeling approach.

***Index Terms*—Computer network performance, differential equations, modeling, transport protocol, wide-area networks.**

I. Introduction

A NEW CLASS of semi-analytical models has recently been introduced in the networking arena, and today appears to be the most promising approach for scalable and accurate performance analysis of large IP networks. These new models, often called *fluid models*, adopt a deterministic description of the average network dynamics through a set of differential equations [1]–[5], thus neglecting the detailed, packet-by-packet description of the stochastic network dynamics. The resulting set of differential equations is then solved numerically, obtaining estimates of the time-dependent network behavior.

The most attractive property of fluid models resides in the fact that their complexity (i.e., the number of differential equations to be solved) is independent of the number of TCP flows and of link capacities, when considering traffic scenarios comprising only long-lived TCP flows (commonly called *elephants*). In addition, fluid models have been recently proved to capture the limiting behavior of TCP elephants in single bottleneck topologies when the number of TCP flows, the bottleneck capacity and the buffer size jointly grow to infinity [1], [6]–[9].

An important limit of the fluid model approaches presented so far in the literature is their poor representation of scenarios comprising the short-lived TCP flows (commonly called *mice*), which are the majority of the flows in the Internet.

Manuscript received April 23, 2004; revised January 24, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Srikant. This work was supported by the Italian Ministry for Education, University and Research, within the TANGO FIRB project. A preliminary version of this paper was presented at the IEEE INFOCOM 2004, Hong Kong.

The authors are with the Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy (e-mail: ajmone@mail.tlc.polito.it; garetto@mail.tlc.polito.it; giaccone@mail.tlc.polito.it; leonardi@mail.tlc.polito.it; schiattarella@mail.tlc.polito.it; tarello@mail.tlc.polito.it).

Digital Object Identifier 10.1109/TNET.2005.860102

In this paper, we develop a fluid model approach in which the sources dynamics are described by *partial* differential equations. This new description of the source dynamics allows the natural representation of TCP mice as well as elephants, with no sacrifice in the scalability of the model. In addition, the use of partial differential equations permits the description of TCP window distributions, instead of averages, thus providing better accuracy in the performance predictions.

The rest of this paper is organized as follows. Section II overviews the fluid model of IP networks originally proposed by Misra, Gong, and Towsley [3]–[5], and Section III discusses other previous works in the same area. Section IV describes the modeling methodology that we propose in this paper, based on partial differential equations, for the case of TCP elephants. We first introduce the simplest version of our model, and then progressively extend it to cope with finite window sizes, fast recovery, and drop-tail buffers. Results are shown along the way, and compared with performance estimates generated by *ns-2* simulations, so as to prove the accuracy of the proposed fluid model approach. In Section V, we move on to considering the case of TCP mice, generalizing the equations adopted for TCP elephants; we point out the fundamental limitation of a deterministic approach to describe the network behavior, and suggest solutions to overcome this limitation. Numerical and simulation results are then presented and discussed for different network scenarios. Finally, Section VI concludes the paper.

II. MGT Fluid Model of IP Networks

In [3]–[5], Misra, Gong, and Towsley presented simple differential equations to describe the behavior of TCP elephants over networks of IP routers adopting a RED (Random Early Detection [10]) active queue management (AQM) scheme. Their approach (that we call MGT) spurred several research efforts aiming at the application of various kinds of fluid models to the performance analysis of packet networks. It is important to note that the equations of the MGT model heavily rely on the assumptions mentioned above (all TCP connections are elephants, and all IP routers adopt RED), and that the extension to mice and drop-tail routers may be not simple.

Consider a network comprising K router output interfaces, equipped with FIFO buffers, and interfacing data channels at rate C (the extension to nonhomogeneous data rates is straightforward). The network is fed by I classes of long-lived TCP flows; all the elephants within the same class follow the same route through the network, thus experiencing the same round-trip time (RTT), and the same average loss probability (ALP).

At time $t = 0$ all buffers are assumed to be empty. Buffers drop packets according to their average occupancy, as dictated by a RED AQM scheme.

A. TCP Source Evolution Equations

Consider the ith class of elephants; the temporal evolution of the average window of TCP sources in the class, $W_i(t)$, is described by the following differential equation:

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(t)} - \frac{W_i(t)}{2}\lambda_i(t) \tag{1}$$

where $R_i(t)$ is the average RTT for class i, and $\lambda_i(t)$ is the loss indicator rate experienced by TCP flows of class i.

The differential equation is obtained by considering the fact that elephants can be assumed to always be in congestion avoidance (CA) mode, so that the window dynamics are close to additive increase, multiplicative decrease (AIMD). The window increase rate in CA mode is approximatively linear, and corresponds to one packet per RTT. The window decrease rate is proportional to the rate with which congestion indications are received by the source, and each congestion indication implies a reduction of the window by a factor two.

B. Network Evolution Equations

$Q_k(t)$ denotes the (fluid) level of the queue in the kth buffer at time t; the temporal evolution of the queue level is described by

$$\frac{dQ_k(t)}{dt} = A_k(t)\left[1 - p_k(t)\right] - D_k(t) \tag{2}$$

where $A_k(t)$ represents the fluid arrival rate at the buffer, $D_k(t)$ the departure rate from the buffer (which equals C, provided that $Q_k(t) > 0$) and the function $p_k(t)$ represents the instantaneous loss probability at the buffer, which depends on the RED parameters. An explicit expression for $p_k(t)$ is given in [3] for RED buffers.

If $T_k(t)$ denotes the instantaneous delay of buffer k at time t, we can write

$$T_k(t) = \frac{Q_k(t)}{C}. \tag{3}$$

If $\mathcal{F}_k$ indicates the set of flows traversing buffer k, $A_k^i(t)$ and $D_k^i(t)$ are respectively the arrival and departure rates at buffer k referred to elephants in class i, it results:

$$A_k(t) = \sum_{i \in \mathcal{F}_k} A_k^i(t)$$

$$\int_0^{t+T_k(t)} D_k^i(a)da = \int_0^t A_k^i(a)\left[1 - p_k(a)\right]da,$$

which means that the total amount of fluid arrived up to time t at the buffer leaves the buffer by time $t + T_k(t)$, since the buffer is FIFO.

C. Source–Network Interactions

Consider elephants in class i. Let $k(h,i)$ be the hth buffer traversed by them along their path P_i of length L_i. The RTT $R_i(t)$ perceived by elephants of class i satisfies the following expression:

$$R_i\left(t + g_i + \sum_{h=1}^{L_i} T_{k(h,i)}(t_{k(h,i)})\right) = g_i + \sum_{h=1}^{L_i} T_{k(h,i)}(t_{k(h,i)}) \tag{4}$$

where g_i is the total propagation delay[1] experienced by elephants in class i, and $t_{k(h,i)}$ is the time when the fluid injected at time t by the TCP sources reaches the hth buffer along its path P_i. We have

$$t_{k(h,i)} = t_{k(h-1,i)} + T_{k(h-1,i)}(t_{k(h-1,i)}). \tag{5}$$

The loss indicator rate is instead given by

$$\lambda_i\left(t + R_i(t)\right) = \alpha\frac{W_i(t)}{R_i(t)}p_i^F(t) \tag{6}$$

where $W_i(t)/R_i(t)$ is the instantaneous emission rate of TCP sources, α is a calibration parameter, and $p_i^F(t)$ is the instantaneous loss probability experienced by elephants in class i:

$$p_i^F(t) = 1 - \prod_{h=1}^{L_i}\left[1 - p_{k(h,i)}(t_{k(h,i)})\right]. \tag{7}$$

Finally

$$A_k(t) = \sum_i \sum_q r_{qk}^i D_q^i(t) + \sum_i e_k^i \frac{W_i(t)}{R_i(t)}N_i \tag{8}$$

where $e_k^i = 1$, if buffer k is the first buffer traversed by elephants of class i, and 0, otherwise; r_{qk}^i is derived by the routing matrix, being $r_{qk}^i = 1$ if buffer k immediately follows buffer q along P_i; N_i is the number of class i active flows.

It can be observed that the MGT fluid model is extremely simple, requiring just one equation per class of elephants, thus being capable of scaling to quite large network models. However, we must also note that the description of TCP mice with the MGT model is not natural, because (obviously) the start time of each mouse determines its window dynamics over time. This aspect is not captured by (1), and one equation has to be written for each mouse, as in [2]. This means that the independence of the fluid model complexity with respect to the number of flows is lost. Moreover, the MGT model, due to the fact that it only describes the average dynamics, also has problems in coping with drop-tail buffers. Finally, the calibration parameter in (6), which is necessary to compensate for the use of the average window size, instead of the window size distribution, must be set empirically.

III. Previous Work on Fluid Models

To the best of our knowledge, fluid models were first proposed in [3] to study the interaction between TCP elephants and a RED buffer in a packet network consisting of just one bottleneck link. In [5], the authors have recently extended their model to

[1]Equation (4) comprises the propagation delay g_i in a single term, as if it were concentrated only at the last hop. This is just for the sake of easier reading, since the inclusion of the propagation delay of each hop would introduce just a formal modification in the recursive equation of $t_{k(h,i)}$.

consider general multi-bottleneck topologies comprising RED routers.

The equations reported in Section II briefly summarize the fluid model proposed in [5], which constitutes the starting point for our work. This set of ordinary differential equations must be solved numerically, using standard discretization techniques.

In [1] and [2], an alternative fluid model has been proposed to describe the dynamics of the average window for TCP elephants traversing a network of drop-tail routers. The behavior of such a network is pulsing: congestion epochs in which some buffers are overloaded (and overflow) are interleaved to periods of time in which no buffer is overloaded, and no loss is experienced, due to the fact that previous losses forced TCP sources to reduce their sending rate. In such a setup, a careful analysis of the average TCP window dynamics at congestion epochs is necessary, whereas sources can be simply assumed to increase their rate at constant speed between congestion epochs. This behavior allows the development of fluid equations and an efficient methodology to solve them. Ingenious queueing theory arguments are exploited to evaluate the loss probability during congestion epochs, and to study the synchronization effect among sources sharing the same bottleneck link. Also in this case, the complexity of the fluid model analysis is independent of link capacities and of the number of TCP flows. An extension that allows considering TCP mice has also been proposed in [1] and [2]. In this case, since the dynamics of TCP mice with different size and/or different start times are different, each mouse must be described with two differential equations; one representing the average window evolution, and one describing the workload evolution. As a consequence, one of the nicest properties of fluid models, the insensitivity of the complexity with respect to the number of TCP flows, is lost.

In [6]–[9], fluid models have been exploited to prove properties related to the asymptotic behavior of a single RED bottleneck topology fed by long lived TCP connections in the so called "many flows regime", i.e., when the number of TCP flows, the bottleneck capacity, and the buffer-size increase jointly to infinity. In particular, in [9] the windows size dynamic for a population of N long-lived TCP connections is represented by a stochastic process whose time samples are the window size distributions (the process is said to be measure-valued), and the mean field dynamics of the process are described by a deterministic "transport equation" which can be approximated by a partial differential equation, under mild assumptions.

In this paper, differently from [9], we adopt an approach which allows us to *directly* obtain an approximate description of the TCP source window size distribution dynamics, based on partial differential equations. We also show that our description of the source dynamics allows a natural representation of mice as well as elephants, with no sacrifice in the scalability of the model.

IV. Modeling TCP Elephants

The class of fluid models that we propose in this paper differs from the previous proposals (with the exception of [9]) because, instead of describing just the evolution of the average window size of TCP sources, we model the evolution of the window size *distribution* for the TCP flow population. This major improvement in the representation of the TCP sources dynamics gives us the advantage of a greater model flexibility, which: 1) allows TCP mice to be described in a way such that the insensitivity of complexity with respect to the number of TCP flows is maintained and 2) permits the modeling of networks in which AQM routers coexist with drop-tail routers.

In other words, rather than just describing the average TCP connection behavior, we statistically model the dynamics of the entire population of TCP flows sharing the same path. This approach leads to systems of partial derivatives differential equations, and produces more flexible models, which scale independently of the number of TCP flows.

In this section, we first introduce the basic model for the TCP flow population. This basic model can be extended by adding several features, which permit a progressively more accurate description of the behavior of TCP sources. Such extensions are described one by one for the sake of readability, but they can be combined at will, to obtain models with the desired level of accuracy and numerical complexity.

A. Basic TCP Sources

To begin, consider a fixed number of TCP elephants. We use $P_i(w,t)$ to indicate the number[2] of elephants of class i whose window is $\leq w$ at time t. For the sake of simplicity, we consider just one class of flows, and omit the index i from all variables. The source dynamics are approximately described by the following equation, for $w \geq 1$:

$$\frac{\partial P(w,t)}{\partial t} = \int_w^{2w} \lambda(\alpha,t)\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha - \frac{1}{R(t)}\frac{\partial P(w,t)}{\partial w} \quad (9)$$

where $\lambda(w,t)$ is the loss indication rate. A formal derivation of (9) is given in Appendix A. Note that this equation is equivalent to the deterministic transport equation reported in Corollary 1 of [9], which was obtained by applying Mean Field Analysis.

The intuitive explanation of the formula is the following. The time evolution of the population described by $P(w,t)$ is governed by two terms: 1) the integral accounts for the growth rate of $P(w,t)$ due to sources with window between w and $2w$ that experience losses, and 2) the second term is the decrease rate of $P(w,t)$ due to sources increasing their window with rate $1/R(t)$.

The quantity $\lambda(w,t)$ can be computed by recalling (6):

$$\lambda(w,t) = \frac{wp^F(t)}{R(t)} \quad (10)$$

in which the current window size of the sources that emitted the lost fluid approximates the window size value at which those sources emitted this fluid. Intuitively, this loss model distributes the lost fluid over the entire population, proportionally to the window size. Note that this loss model does not require any calibration parameter, contrary to the MGT model; indeed, statistics like the variance of the TCP flow windows size impacts on the network stationary behavior.

[2] $P_i(w,t)$ is assumed to be a continuous function $\mathbb{R}^2 \rightarrow \mathbb{R}$ due to the fluid nature of the model.

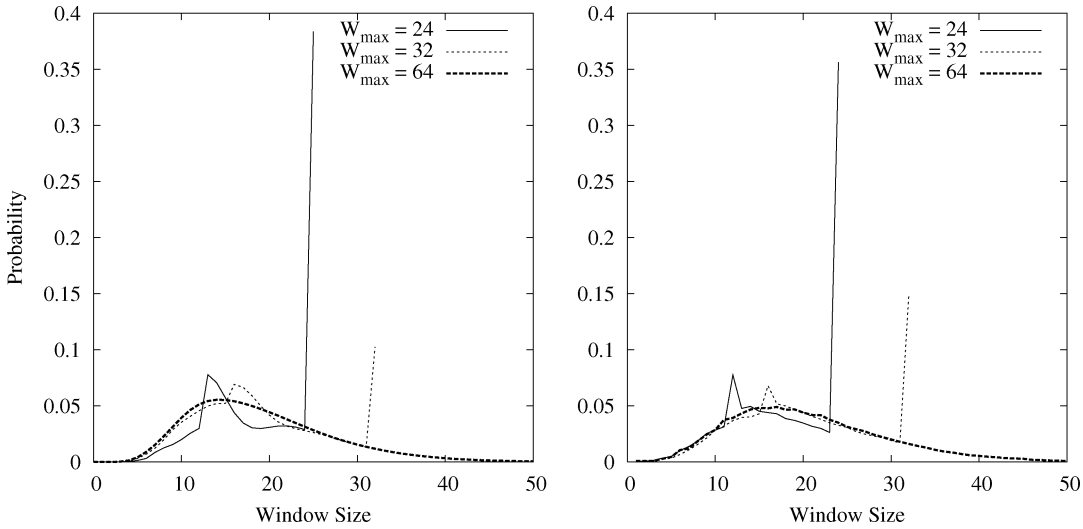

Fig. 1. Fluid model (left) and *ns-2* (right): average window size distribution for 8 TCP elephants traversing a single bottleneck link with RED buffer, varying their maximum window size; these TCP flows compete with 8 other TCP elephants with maximum window size 64.

B. Accounting for the Maximum Window Size

We now extend the basic model of (9) to account for the maximum window size of TCP sources, that we denote by $W^{\max}$. It holds that

$$\frac{\partial P(w,t)}{\partial t} = \int_{w}^{\min(2w,W^{\max})} \lambda(\alpha,t)\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha + \lambda(W^{\max},t)P_{\max}(t)u\left(w-\frac{W^{\max}}{2}\right) - \frac{1}{R(t)}\frac{\partial P(w,t)}{\partial w} \quad (11)$$

for $1 \le w < W^{\max}$, where $u(\cdot)$ is the unit step function, and $P_{\max}(t)$ is the number of TCP flows whose window is exactly equal to $W^{\max}$.

For $P_{\max}(t)$ we can write

$$\frac{dP_{\max}(t)}{dt} = \frac{1}{R(t)}\lim_{w\uparrow W^{\max}}\frac{\partial P(w,t)}{\partial w} - \lambda(W^{\max},t)P_{\max}(t) \quad (12)$$

with the boundary conditions $P(1^{-},t) = 0$ and $\lim_{w\uparrow W^{\max}} P(w,t) + P_{\max}(t) = N$. The derivation of (11) is very similar to that of (9). The first term in (11) is the contribution of all TCP sources which experience losses at window size between w and $2w$ ($W^{\max}$ if $2w$ exceeds it). The second term of (11) is the contribution of all TCP sources at maximum window size that experience losses; note that this contribution exists only for windows greater than $W^{\max}/2$.

The growth rate of $P_{\max}(t)$ is obtained as the limit of the usual growth rate $(\partial P(w,t)/\partial w)/R(t)$ of $P(w,t)$. The decrease rate of $P_{\max}(t)$ is simply $\lambda(W^{\max},t)$.

C. Experiments With RED Buffers

In this subsection, we discuss some numerical results referring to the mathematical model in (11). Before proceeding, we notice that all the results shown in this paper were obtained by numerically solving the model. For this purpose, we applied standard discretization techniques; in particular, a first-order finite differences method for the sources equations, and a fourth-order Runge–Kutta method for the queue equations, as better explained in Appendix E.

TABLE I
MAXIMUM WINDOW SIZE $W^{\max}$ AND AVERAGE WINDOW SIZE (AWS) (IN PACKETS) FOR CLASS 2 FLOWS, AVERAGE QUEUE LENGTH (AQL) (IN PACKETS) AND AVERAGE LOSS PROBABILITY (ALP) FOR THE EXPERIMENTS OF SECTION IV-C

W^{max}	Fluid model			ns		
	AQL	ALP	AWS	AQL	ALP	AWS
24	18.7	0.0037	18.3	18.2	0.0029	18.3
32	19.8	0.0042	19.7	18.9	0.0032	20.2
64	20.2	0.0044	20.2	19.2	0.0034	20.6

Consider the case of a single bottleneck link topology in which a gentle version of the RED AQM algorithm ($\min\_th = 10$, $\max\_th = 160$, $p\_\max = 0.1$, $w = 0.0001$) is implemented, with two classes of eight TCP elephants saturating the link capacity ($C = 100$ Mb/s), assuming a propagation delay equal to 30 ms. We compare the results of three different experiments, in which the first elephant class (class 1) has always maximum window size 64, while the other class (class 2) has maximum window size 64, 32, and 24. The packet size for this and all other experiments in this paper is 10 000 bits. In Fig. 1, we show the window size probability density function of elephants in class 2 predicted by our model and by *ns-2*. In Table I, we compare the average window size, the average queue length and the loss probability for the model and the *ns-2* simulator. Note that for lower $W^{\max}$, the average window size of class 2 elephants is smaller; at the same time, the average window size for class 1 flows increases, so that the average window size of all the 16 TCP elephants is roughly constant and equal to 20. A model without window size clipping, like for example the one in [3]–[5], is capable of correctly estimating the average window size of the 16 elephants, but fails in capturing the differences among classes with different maximum window size values. The comparison clearly shows that the fluid model is quite accurate.

D. Considering Fast Recovery

Newer versions of TCP (such as NewReno—see RFCs 2581 and 3782) avoid halving the window more than once for RTT,

even in the case of multiple losses. To model this fact, we divide the population of TCP flows whose congestion window is $\leq w$ at time t in two classes: class L comprises all sources that experienced losses during the last RTT, while class O is composed by remaining sources,[3] so that $P(w,t) = P_L(w,t) + P_O(w,t)$.

We can write

$$\frac{\partial P_O(w,t)}{\partial t} = -\int_1^w \lambda(\alpha,t)\frac{\partial P_O(\alpha,t)}{\partial \alpha}d\alpha - \frac{1}{R(t)}\frac{\partial P_O(w,t)}{\partial w} + \frac{1}{R(t)}P_L(w,t) \quad (13)$$

$$\frac{\partial P_L(w,t)}{\partial t} = +\int_1^{2w} \lambda(\alpha,t)\frac{\partial P_O(\alpha,t)}{\partial \alpha}d\alpha - \frac{1}{R(t)}P_L(w,t) - \frac{1}{R(t)}\frac{\partial P_L(w,t)}{\partial w}. \quad (14)$$

A formal derivation of (13) and (14) is reported in Appendix B. An intuitive explanation of the two equations can be provided as follows. In the right-hand side of (13), the first two terms account for the decrease rate of the number of elephants of class O whose window is $\leq w$ at time t, due to: (i) sources in class O experiencing losses and moving to class L, (ii) sources in class O increasing their window. The third term refers to the sources moving to class O from class L after experiencing a RTT without losses. In the right-hand side of (14), the first term accounts for the growth rate of the number of elephants of class L whose window is $\leq w$ at time t, due to sources in class O experiencing losses. The second and third terms account for the decrease rate due to: 1) sources moving to class O from class L after a RTT without losses and 2) sources in class L increasing their window.

More general fluid equations describing TCP elephants and accounting for the TCP threshold mechanisms and for time-outs are reported in [13].

E. Modeling Drop-Tail Buffers

As we have already mentioned, a fluid model for the description of RED AQM schemes was originally proposed in [3]. RED matches quite well the fluid modeling approach, since in RED buffers the loss probability is a smooth function of the queue length averaged over a rather long time window. The case of drop-tail buffers is instead much more difficult to describe with fluid models, since in this case the loss probability is a discontinuous function of the instantaneous queue size.

Many studies have shown that the behavior of networks carrying TCP traffic is pulsing: congestion epochs in which some buffers are overloaded (and overflow) are interleaved to periods of time in which traffic is lighter, buffers are not saturated, and no loss is experienced. Light traffic periods are the result of losses at the previous congestion epochs, that force TCP sources to reduce their emission rate. As a consequence, the loss processes experienced by TCP flows traversing drop-tail buffers are quite bursty. This burstiness induces a high degree of correlation (synchronization) among the dynamics of TCP sources sharing the same buffer. In addition, during congestion epochs, losses are not evenly distributed among TCP flows, but are more likely to affect TCP sources with larger window size. In this context, it is necessary to distinguish among sources with different instantaneous window size, while at the same time accounting for the effects of the TCP fast recovery mechanism, which prevents TCP sources from halving their window several times within one round trip time.

[3]For the sake of simplicity, the equations in this section and in the rest of the paper do not consider the effect of the maximum window size. However, in all numerical results that are presented in this paper the effect of the maximum window size is always accounted for.

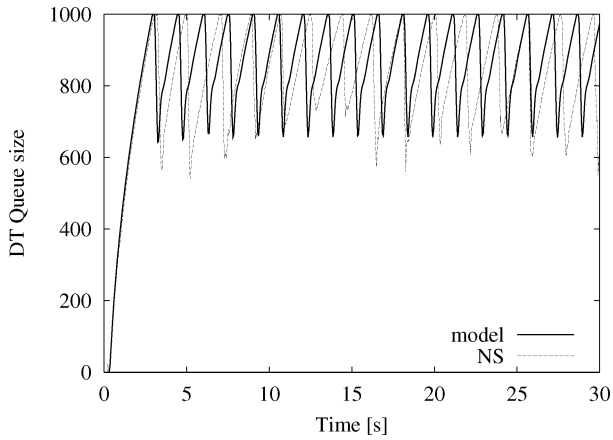

Fig. 2. Fluid model and *ns-2* simulator: queue size evolution for one bottleneck link fed by a drop-tail buffer and traversed by TCP elephants.

The level of detail in the description of the TCP sources dynamics adopted in this paper allows an easy description of the time-dependent behavior of the packet loss probability:

$$p_k(t) = \frac{\max(0, A_k(t) - C)}{A_k(t)} \mathbb{1}_{\{Q_k(t)=B_k\}} \quad (15)$$

that is, the loss probability $p_k(t)$ equals $(A_k(t) - C)/A_k(t)$ (the relative difference between the instantaneous arrival rate and the service rate) only when the buffer is full, being B_k the capacity of buffer k, and $\mathbb{1}_{\{.\}}$ the indicator function.

A different approach is used in [1] and [2] to describe the dynamics of the average window size for TCP flows traversing a network with drop-tail buffers. In those papers, the loss indicator rate is obtained by applying queueing theory results which are not "internal" to the fluid model. That approach is probably difficult to generalize to networks including both drop-tail and AQM buffers.

F. Experiments With Drop-Tail Buffers

In this subsection, we briefly comment some numerical results obtained with our modeling approach in the case of drop-tail buffers.

First, we consider the case of a single bottleneck link (with data rate $C = 100$ Mb/s, propagation delay 30 ms), traversed by just one class of 30 TCP elephants, with maximum window size 64 packets; the maximum buffer size is set to 1000. The curves in Fig. 2 show the queue size evolution over time. Our model captures the well-known oscillating behavior of TCP, which was observed in simulation experiments as well as measurements [11], [12].

The results of *ns-2* simulations are reported in Fig. 2 and Table II for comparison, and again show that the fluid model is accurate.

TABLE II
AVERAGE WINDOW SIZE (AWS), AVERAGE QUEUE LENGTH (AQL) AND AVERAGE LOSS PROBABILITY (ALP) FOR THE SETUP OF SECTION IV-F

	AWS	AQL	ALP
Fluid model	39	833	0.0013
ns-2	38.4	831	0.0013

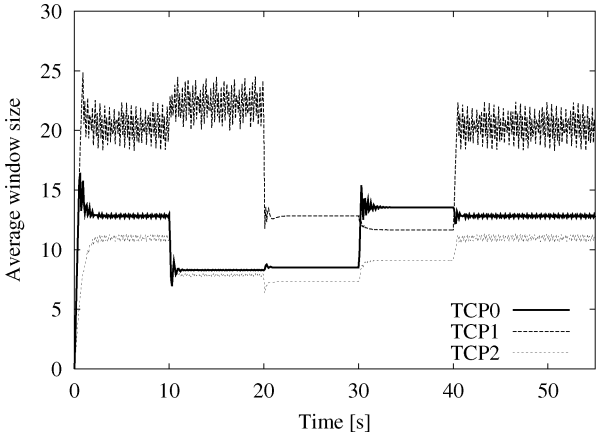

Fig. 3. Fluid model: window size evolution for three long-lived TCP flows with interfering UDP traffic.

The second scenario we consider is a network topology comprising two links, the first fed by a RED buffer, the second fed by a drop-tail buffer.[4] The links are crossed by five classes of elephants. Two classes of TCP flows are single-hop (TCP0 crosses the first link, TCP1 crosses the second one), while the other one (TCP2) crosses both links; the two links are also crossed by two interfering classes of CBR UDP flows (UDP3 crosses the RED buffer, UDP4 crosses the drop-tail buffer). UDP3 is on in the time interval [10, 30] s, UDP4 in the time interval [20, 40] s: when the UDP flows are on, they consume about 40% of the bandwidth of their link. Fig. 3 shows plots of the window size evolution for the three TCP flow classes. When UDP3 starts, the window size of the two TCP flow classes sharing the same link decreases; when also UDP4 starts, the window size of TCP1 decreases, and again that of TCP2 goes down, in favor of TCP0. The window size of TCP0 and TCP2 increases when UDP3 ends, while those of TCP1 and TCP2 increase when UDP4 ends.

The results of *ns-2* simulations for the same setup once more show that the fluid model is quite accurate: for instance, in Fig. 4 we overlap the curves of the model and the *ns-2* simulator for the TCP0 elephants. Complete *ns-2* results can be found in [13].

These results prove that our model can cope with both controlled (TCP) and uncontrolled (UDP) long-lived flows, and is capable of predicting the TCP transient effects due to the presence of on-off interfering sources.

V. MODELING TCP MICE

We now come to the very important issue of modeling TCP mice, whose dynamics are mostly, if not completely, due to the slow-start algorithm, and in particular to the first slow-start phase that is executed when the TCP connection is opened. For this reason, in order to model TCP mice, we model the initial slow-start phase up to the first loss or to the first hit of the maximum window size, and then we assume that flows stay in congestion avoidance for the rest of the connection lifetime.

[4]It is worth observing that all previous applications of fluid models to packet networks always considered either RED buffers, or drop-tail buffers, but the two types of buffers were never mixed, since the fluid models could not support this feature.

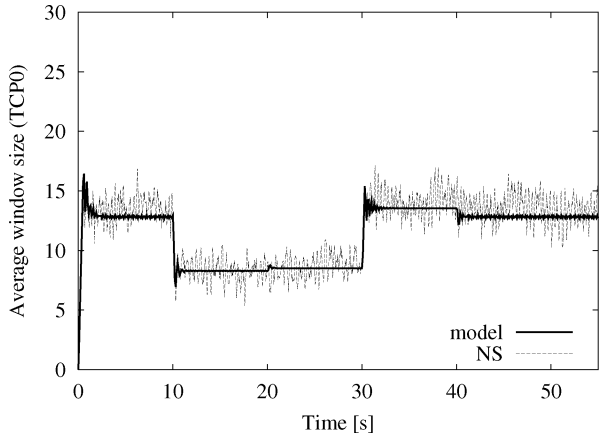

Fig. 4. Overlap of TCP0 curves from the fluid model (Fig. 3) and *ns-2*.

Let $P_s(w,t,l)$ be the number of flows in slow-start with window size $\leq w$ and residual workload $\leq l$ at time t. Analogously, $P(w,t,l)$ refers to flows in congestion avoidance. We can write

$$\begin{aligned}\frac{\partial P(w,t,l)}{\partial t} = &-\frac{1}{R(t)}\frac{\partial P(w,t,l)}{\partial w}-\frac{w}{R(t)}\frac{\partial P(w,t,l)}{\partial l}\bigg|_{l=0}\\ &+\frac{w}{R(t)}\frac{\partial P(w,t,l)}{\partial l}+\int_w^{2w}\lambda(\alpha,t)\frac{\partial P(\alpha,t,l-1)}{\partial \alpha}d\alpha\\ &+\int_1^{2w}\lambda(\alpha,t)\frac{\partial P_s(\alpha,t,l-1)}{\partial \alpha}d\alpha \qquad (16)\\ \frac{\partial P_s(w,t,l)}{\partial t} = &-\frac{w}{R(t)}\frac{\partial P_s(w,t,l)}{\partial w}\\ &-\frac{w}{R(t)}\frac{\partial P_s(w,t,l)}{\partial l}\bigg|_{l=0}+\frac{w}{R(t)}\frac{\partial P_s(w,t,l)}{\partial l}\\ &-\int_1^{w}\lambda(\alpha,t)\frac{\partial P_s(\alpha,t,l-1)}{\partial \alpha}d\alpha+\gamma(t,l). \qquad (17)\end{aligned}$$

A formal proof of these equations is given in Appendix C. An intuitive explanation is as follows. In (16), the first two terms on the right-hand side account for the decrease rate of $P(w,t,l)$ due to: 1) sources increasing their rate (first term) and 2) sources terminating because of null residual workload (second term). The last three terms account for the growth rate of $P(w,t,l)$. The third term takes into account those sources with previous residual workload slightly greater than l, assuming at time t a value $\leq l$; the weight of this term is $w/R(T)$ because the workload is reduced with an average rate equal to w units of fluid per RTT. The fourth term represents those sources in congestion avoidance with window between w and $2w$ and residual workload $\leq l-1$ that experience a loss. They are added to $P(w,t,l)$ because their window is halved (and becomes $\leq w$) and their residual workload goes back to l, as the lost unit of fluid must be retransmitted. Finally, the fifth term represents an increase similar to the fourth term, applied to sources in slow-start: these sources, with window size between 1 and $2w$ and residual workload $\leq l-1$, experience a loss and consequently move to a state

in which they are in congestion avoidance, their window is $\leq w$ and their residual workload goes back to l.

Equation (17) is very similar to (16), since the evolution of $P_s(w,t,l)$ with respect to the residual workload (second and third terms) is the same, and the first term differs only for the fact that the window growth is in this case exponential rather than linear. Moreover, the fourth term refers to sources moving into congestion avoidance because of a loss [similarly to the fifth term of (16)]. Finally, being $\gamma(t)$ the mice arrival rate, the last term accounts for newly activated TCP mice.

Note that the representation of the TCP window dynamics over the (t,w) space allows us to distinguish among TCP mice with different instantaneous window sizes, thus providing the correct level of detail for the analysis of this type of TCP flows. Indeed, TCP mice open in slow-start, with window 1, and then their window evolves according to (16) and (17).

The model of TCP mice can be simplified by assuming flow lengths to be exponentially distributed, with average L. Thanks to the memoryless property of the exponential distribution, we can write

$$\begin{aligned}\frac{\partial P(w,t)}{\partial t} = & -\frac{1}{R(t)}\frac{\partial P(w,t)}{\partial w} \\ & -\frac{(1-\bar{p}_L(t))}{R(t)L}\int_1^w \alpha\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha \\ & +\int_w^{2w}\lambda(\alpha,t)\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha \\ & +\int_1^{2w}\lambda(\alpha,t)\frac{\partial P_s(\alpha,t)}{\partial \alpha}d\alpha \qquad (18)\end{aligned}$$

$$\begin{aligned}\frac{\partial P_s(w,t)}{\partial t} = & -\frac{w}{R(t)}\frac{\partial P_s(w,t)}{\partial w} \\ & -\frac{(1-\bar{p}_L(t))}{R(t)L}\int_1^w \alpha\frac{\partial P_s(\alpha,t)}{\partial \alpha}d\alpha \\ & -\int_1^w \lambda(\alpha,t)\frac{\partial P_s(\alpha,t)}{\partial \alpha}d\alpha+\gamma(t) \qquad (19)\end{aligned}$$

where $\bar{p}_L(t)$ is the average loss probability experienced by the flow, during its total active period. The formal derivation of the second term is reported in Appendix D. We can approximate $\bar{p}_L(t)$ by using the same approach proposed in [3] and [4] to evaluate the average loss probability in a RED queue; we obtain

$$\frac{\partial \bar{p}_L(t)}{\partial t} = -\frac{W(t)}{LR(t)}\bar{p}_L(t)+\frac{W(t)}{LR(t)}p^F(t) \qquad (20)$$

with $p^F(t)$ the instantaneous loss probability, defined in (6), and $W(t)$ the average window size at time t.

We wish to stress the fact that (18)–(20) provide quite a powerful tool for an efficient representation of TCP mice, since a wide range of distributions (including those incorporating long-tail distributions) can be approximated with an arbitrary degree of accuracy by a mixture of exponential distributions [14].

A. Randomness in Fluid Models

The fluid models that we have presented so far provide a deterministic description of the network behavior, thus departing from the common approach of attempting a probabilistic description of the network dynamics by means of stochastic models, such as continuous-time or discrete-time Markov chains and queueing models.

Deterministic fluid models have been proven to represent correctly the asymptotic behavior of TCP when the number of active flows (elephants) tends to infinity [8].

Indeed, when considering scenarios with only elephants, randomness, which is completely lost in fluid models, plays a minor role, because queues tend to be heavily congested, and the loss rate is basically determined by the load offered by TCP connections in excess of the bottleneck capacity.

Instead, fluid models are not suitable to analyze network scenarios in which the capacity of the links is not saturated. In particular, they completely fail to predict the behavior of a network loaded only by TCP mice: if links are underloaded (i.e., their average utilization is smaller than one), fluid models predict that buffers are constantly empty. This is not what we observe and measure in packet networks; the discrepancy is essentially due to the fact that, in underload conditions, the stochastic nature of the input traffic plays a fundamental role that cannot be neglected.

Randomness, indeed, impacts the system behavior at many levels:

1) at flow level, since the arrival process of TCP flows exhibits a nonnegligible burstiness, which causes the short-term offered load at the queues to randomly vary over time, thus leading to sporadic periods of congestion;
2) at packet level, since the arrival process of packets at queues exhibits a bursty behavior, thus causing sporadic buffer overloads also during periods in which the average utilization factor is smaller than 1.

This implies that, when analyzing the behavior of underloaded networks, the complete determinism of fluid models is not satisfactory: we must stop short of reducing the network operations to the deterministic evolution of average parameters, keeping in the model some of the stochastic characteristics of the network behavior. This can be done by using stochastic differential equations, rather than deterministic differential equations, to describe the system dynamics, and then solving them (for example) with a Monte Carlo technique.

In practice, with respect to the model we just presented, we can:

1) use a Poisson counter with average $\gamma(t)$ to describe the mice arrival rate, rather than a deterministic rate, so that instead of (19) we have a stochastic partial differential equation.
2) use a nonhomogeneous Poisson process to describe the completion of TCP connections; the average at time t of such process is represented by the second term of (18); by so doing, also (18) becomes a stochastic partial differential equation. Note that this suggests that, due to retransmissions, the completion time of connections increases as their loss probability grows.
3) use a Poisson point process (possibly with batch arrivals) to describe the workload emitted by TCP sources, rather than a continuous deterministic fluid process, but keeping the average rate $W_i(t)N_i/R_i(t)$.

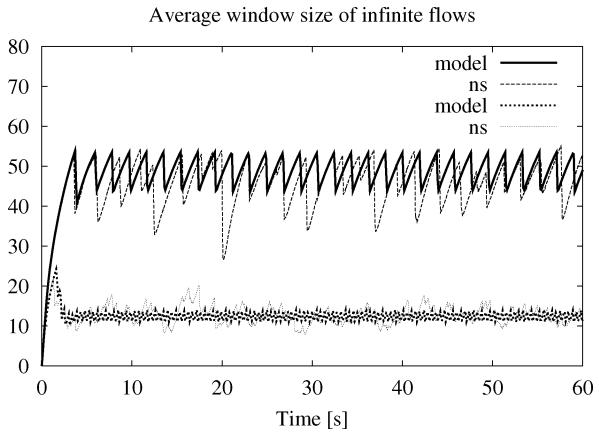

Fig. 5. Average window size evolution for elephants competing with mice on a single bottleneck link; the average window size decreases when the mice arrival rate grows from 100 to 400 connections/s.

Of course, this is only one possible approach to account for randomness when studying the behavior of TCP mice; we do not claim any optimality of this approach, and a deeper investigation is needed about the possible ways of coping with randomness, without losing the property of independence of the model complexity with respect to the number of flows.

B. Experiments With Mice

In this subsection, we discuss results for network scenarios comprising TCP mice. First, we consider a case in which both mice and elephants coexist. Second, we investigate the impact of the source emission model in a scenario where only mice are active. Third, we study the impact of the flow size distribution. Finally, we investigate the invariance properties of the network when mice are present.

1) Results With Both Elephants and Mice: These first results refer to a single bottleneck link fed by a drop-tail buffer. The buffer size is equal to 1000 packets, the link capacity is $C = 100$ Mb/s, the propagation delay between TCP sources and buffer is 30 ms. Twenty TCP elephants are active, with maximum window size 64 packets, and coexist with TCP mice, whose length is geometrically distributed with mean 20 segments. The TCP mice arrival rate is set equal to 100, 200, and 400 connections per second. The presence of elephants is crucial in order to saturate the link bandwidth, because they consume the capacity that is not used by mice. Indeed, in Fig. 5 and Table III we can see that the average window size for elephants decreases when the arrival rate of mice increases. In the same table, we also report the average completion time (ACT) of mice, obtained from the average number of active mice, by applying Little's theorem.

Table III and Fig. 5 also report the results of *ns-2* simulations for the same setup, for comparison: the fluid model can be observed to be quite accurate in this case too.

2) Results With Mice Only: Impact of the Emission Model: If elephants are removed from the network, deterministic fluid models do not provide useful information about the network behavior, as explained in Section V-A. As a consequence, we now consider fluid models employing the stochastic extensions described in Section V-A.

TABLE III
ARRIVAL RATES (AR), AVERAGE COMPLETION TIMES (ACT), AVERAGE WINDOW SIZE (AWS), AVERAGE QUEUE LENGTH (AQL) AND AVERAGE LOSS PROBABILITIES (ALP) FOR THE EXPERIMENTS OF SECTION V-B.

	mice		elephants	bottleneck	
	AR (flows/s)	ACT (ms)	AWS (pck)	AQL (pck)	ALP
Fluid model	100	498	48.9	887	0.0006
	200	510	37.4	913	0.0014
	400	537	12.4	893	0.015
ns-2	100	508	45.8	806	0.0027
	200	512	34.9	806	0.0054
	400	750	12.9	926	0.024

TABLE IV
PARAMETERS OF THE HYPER-EXPONENTIAL DISTRIBUTION APPROXIMATING THE PARETO DISTRIBUTION

Prob.	mean length
$7.88\ 10^{-1}$	6.48
$1.65\ 10^{-1}$	23.26
$3.70\ 10^{-2}$	80.65
$8.34\ 10^{-3}$	279.7
$1.87\ 10^{-3}$	970.2
$4.22\ 10^{-4}$	3376
$9.46\ 10^{-5}$	11862
$2.10\ 10^{-5}$	43086
$4.52\ 10^{-6}$	176198

Consider a single bottleneck link fed by a drop-tail buffer, with capacity equal to 1000 packets. The link data rate C is 1.0 Gb/s, while the propagation delay between TCP sources and buffer is 30 ms. In order to reproduce a TCP traffic load close to what has been observed on the Internet, flow sizes are distributed according to a Pareto distribution with shape parameter equal to 1.2 and scale parameter equal to 4.

Using the algorithm proposed in [14], we approximated the Pareto distribution with a hyper-exponential distribution of the ninth order, whose parameters are reported in Table IV. The resulting average flow length is 20.32 packets. Correspondingly, nine classes of TCP mice are considered in our model. The maximum window size is set to 64 packets for all TCP sources. Experiments with loads equal to 0.6, 0.8 and 0.9 were run; however, for the sake of brevity, we report here only the results for load equal to 0.9.

Fig. 6 compares the queue lengths distributions obtained with *ns-2*, and with the stochastic fluid model.

While in the model the flow arrival and completion processes have been randomized according to a nonhomogeneous Poisson process (see Section V-A), different approaches have been considered to model the traffic emitted by sources in a small interval $[t, t + \Delta t)$:

Poisson: the emitted traffic is a Poisson process with time-varying rate;

Det-B: the emitted traffic is a batch Poisson process with time-varying rate and constant batch size, equal to the instantaneous average TCP mice window size;

Exp-B: the emitted traffic is a batch Poisson process with time-varying rate and exponential batch size, whose mean is equal to the instantaneous average TCP mice window size;

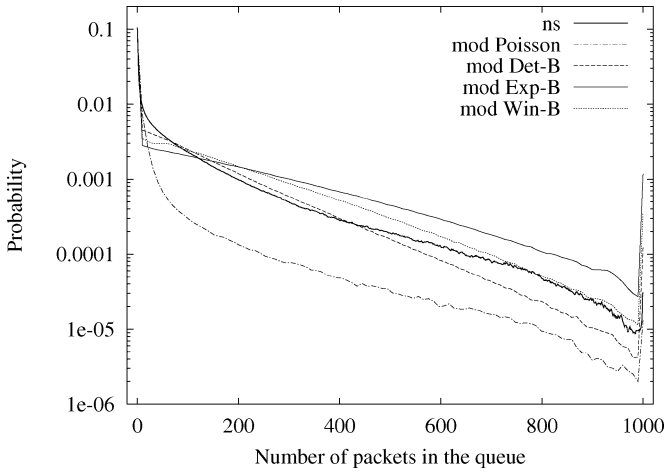

Fig. 6. Queue length distribution for single drop-tail bottleneck, varying the random process modeling the workload emitted by the TCP sources; comparison with *ns-2* simulator.

TABLE V
AVERAGE LOSS PROBABILITY (ALP), AVERAGE QUEUE LENGTH (AQL) AND AVERAGE COMPLETION TIMES (ACT) IN SECONDS OF THE NINE CLASSES OF MICE FOR THE SETUP OF SECTION V-B

	ALP	AQL	ACT[s]
Poisson	$1.23\ 10^{-6}$	17.61	0.0932, 0.129, 0.169 0.274, 0.613, 1.68 5.48, 19.2, 73.7
Win-B	$1.22\ 10^{-4}$	143.12	0.0991, 0.138, 0.187 0.297, 0.658, 1.92 6.29, 22.4, 95.1
ns-2	$5.34\ 10^{-5}$	101.63	0.104, 0.160, 0.219 0.327, 0.661, 1.83 6.10, 21.3, 87.0

TABLE VI
PARAMETERS OF THE THREE FLOW LENGTH DISTRIBUTIONS

	σ	mean length 1	mean length 2
Distr. 1	20.32	20.32	-
Distr. 2	28.89	6.48	80.65
Distr. 3	215.51	6.48	3376.24

Win-B: the emitted traffic is a batch Poisson process with time-varying rate, in which the batch size distribution is equal to the instantaneous TCP mice window size distribution.

The Poisson approach corresponds to the most traditional and simple choice; however, as shown in [15], the traffic emitted by a population of TCP sources, along with the well-known effects of long-range correlation, which are essentially due to the slow fluctuations in the number of active flows (as shown later), also exhibits some short-term burstiness, which is intimately related to the TCP window mechanism, and must be considered by an accurate model. For this reason, following the approach proposed in [15], we modeled the traffic emitted by TCP sources as a Poisson process with batch arrivals, where the batch size is adapted to the current windows size of TCP sources. According to the *Det-B* and *Exp-B* approaches, the batch size distribution is adapted to the current TCP window size distribution by matching just the first moment; instead, with the *Win-B* approach, a complete match between the batch size and the window size distributions is possible. As a consequence, we expect that the *Win-B* approach outperforms both *Det-B* and *Exp-B*. Indeed, Fig. 6 confirms our expectations. If we use a Poisson process to model the instants in which packets (or, more precisely, units of fluid) are emitted by TCP sources, the results generated by the fluid model cannot match the results obtained with the *ns-2* simulator. Instead, the performance predictions obtained with the fluid model become quite accurate when the workload emitted by TCP sources is taken to be a Poisson process with batch arrivals. The best fitting (confirmed also by several other experiments, not reported here for lack of space) is obtained for batch size distribution equal to the instantaneous TCP mice window size distribution (case *Win-B*). Note that our proposed class of fluid models naturally provides the information about the window size distribution.

Table V reports the average loss probability, the average queue length, and the average completion time for each class of TCP mice, obtained either with *ns-2*, or with the *Poisson* and *Win-B* models. The *Poisson* model significantly underestimates the average queue length and loss probability, thus producing an optimistic prediction of completion times. The *Win-B* model moderately overestimates the average queue length and loss probability, as pointed out in [15]. However, for very short flows, completion time predictions obtained with the *Win-B* model are slightly optimistic; this is mainly due to the fact that an idealized TCP behavior (in particular, without timeouts) is considered in the model.

3) Results With Mice Only: Impact of the Flow Size: We now discuss the ability of our model to capture the impact on the network behavior of the flow size variance.

We consider three different scenarios, in which flow lengths are distributed according to either an exponential distribution ("Distr.1"), or hyper-exponentials of the second order ("Distr.2" and "Distr.3"). For all three scenarios, we keep the average flow size equal to 20.32 (this is the average flow size used in the previous subsection), and we vary the standard deviation σ. Detailed parameters of our experiments are reported in Table VI.

Table VII shows a comparison between the results obtained with either the *Win-B* model or *ns-2*. As in previous experiments, the model moderately overestimates both the average loss probability and the average queue length. The discrepancies in the average completion times between model and *ns-2* remain within 10%.

Fig. 7, which reports the queue length distributions obtained by the model in the three scenarios, emphasizes the significant dependency of the queue behavior on the flow size variance. This dependency is mainly due to the complex interactions between the packet-level and flow-level dynamics which are due to the TCP protocol.

4) Results With Mice Only: Impact of the Link Capacity: Finally, we discuss the effect on performance of the link capacity. The objective of this last study of networks loaded with TCP mice only is to verify whether the performance of networks which differ for a multiplicative factor in capacities show some type of invariance, like in the case of elephants.

More precisely, we wish to determine whether the queue length distribution exhibits any insensitivity with respect to the bottleneck link capacity, for the same value of the traffic

TABLE VII
AVERAGE LOSS PROBABILITY (ALP), AVERAGE QUEUE LENGTH (AQL) AND AVERAGE COMPLETION TIMES (ACT) IN SECONDS OF THE DIFFERENT CLASSES OF MICE FOR THE SETUP OF SECTION V-B, HAVING INTRODUCED RANDOM ELEMENTS

	ALP	AQL	ACT[s]
Distr.1(model)	0.0	63.9	0.131
Distr.1(ns-2)	0.0	40.9	0.128
Distr.2(model)	$4.01\ 10^{-5}$	123	0.0985,0.185
Distr.2(ns-2)	$9.00\ 10^{-6}$	98.0	0.0878,0.191
Distr.3(model)	$3.29\ 10^{-4}$	167	0.0999,2.01
Distr.3(ns-2)	$1.23\ 10^{-4}$	142	0.0915,1.85

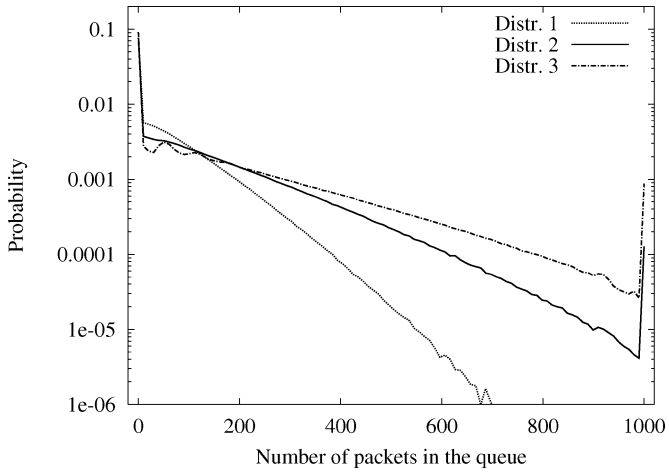

Fig. 7. Queue size distribution for single drop-tail bottleneck, varying the flow length distribution.

intensity. This curiosity is motivated by the fact that in many classical queueing models (e.g., the M/M/1 queue, possibly with batch arrivals) the queue length distribution depends only on the average load, not on the server speed.

We consider the third scenario ("Distr.3") of the previous experiment, we fix the traffic load at 0.9, and we study four different networks, in which the bottleneck capacity is equal to 10 Mb/s, 100 Mb/s, 1 Gb/s, and 10 Gb/s, respectively.

The results of the fluid model, depicted in Fig. 8, show that, in general, the queue length distribution exhibits a dependency on the link capacity. The packet-level behavior, indeed, strongly depends on flow-level dynamics, which cause a slowly varying modulation of the arrival rate at the packet level. The flow-level dynamics, however, do not scale up with the capacity of the system, since the random variable which represent the number of active flows has a coefficient of variation which decreases as we increase the system capacity (consider, for example, the Poisson distribution of the number of active flows proposed in [16]).

Nevertheless, when the capacity of the system becomes very large (in the considered example, greater than 1 Gb/s) the dependence of the queue distribution on capacity tends to vanish, and the queueing behavior becomes indeed independent from the link capacity. This phenomenon was confirmed by *ns-2* simulations.

This behavior is mainly due to the fact that when the capacity becomes very large, the coefficient of variation of the number of active flows becomes small. As a consequence, the effects of

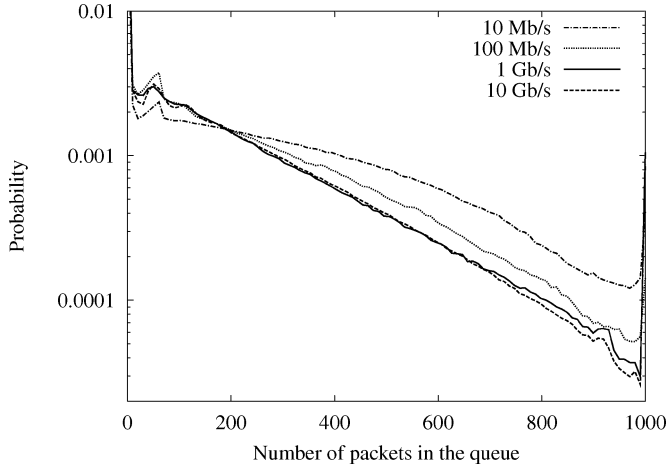

Fig. 8. Queue size distribution for single drop-tail bottleneck, varying the bottleneck capacity.

the flow-level dynamics on the network performance tend to become negligible, and the packet-level behavior resembles that of a single server queue loaded by a stationary Poisson (or batched Poisson) process, for which the queue length distribution is independent of the server capacity.

To confirm this intuition, we solved the fluid model by eliminating the randomness at the flow level (i.e., in the flow arrival and departure processes), and we observed that the dependency on the capacity disappears.

We would like to remark, however, that the flow length distribution plays a major role in determining the system capacity above which the queue length distribution no longer depends on the system capacity—the invariance phenomenon appears at higher data rates when the variance of the flow length distribution increases.

VI. CONCLUSION

In this paper, we have proposed a new fluid model approach for the investigation of the performance of IP networks loaded by TCP mice and elephants (as well as UDP flows). Our approach exploits *partial* differential equations, thus permitting the description of distributions, instead of averages, hence achieving better accuracy in the results with respect to previously proposed fluid modeling approaches.

The performance estimates obtained with our fluid models have been compared against *ns-2* simulations in the cases in which the latter are feasible, proving both the accuracy and the scalability of the proposed modeling approach.

In case of underloaded networks populated only by TCP mice, we have pointed out a fundamental limitation in the deterministic approach to describe the network dynamics, and we have suggested and discussed different solutions to introduce randomness in fluid models in order to obtain reliable predictions of the system behavior.

APPENDIX A
PROOF OF (9)—BASIC SOURCES

We wish to estimate the evolution of $P(w,t)$; we define $v(w,t) = \partial P(w,t)/\partial w$ as the probability density of the

window distribution at time t. Consider a small enough Δt such that $R(t) \approx R(t+\Delta t)$. Let ΔP^- be the number of sources with window $\leq w$ at time t, but with window $> w$ at time $t+\Delta t$. All the sources which do not experience any loss indication during the interval $[t, t+\Delta t)$ increase their window with rate $1/R(t)$. Among these sources, ΔP^- includes only the ones with initial window $> w - \Delta t/R(t)$, since they will exceed w by time $t+\Delta t$. If we assume that the loss indication process can be approximated locally (i.e, in the small interval $[t, t+\Delta t)$) with a Poisson process with rate $\lambda(w,t)$, the probability that no losses are experienced during Δt is $(1-\lambda(\alpha,t)\Delta t + o(\Delta t))$; then

$$\Delta P^- = \int_{w-\Delta t/R(t)}^{w} (1-\lambda(\alpha,t)\Delta t + o(\Delta t))\, v(\alpha,t) d\alpha$$
$$\frac{\Delta P^-}{\Delta t} \to \frac{1}{R(t)} v(w,t). \tag{21}$$

Now let ΔP^+ be the number of sources with window $> w$ at time t, but with window $\leq w$ at time $t+\Delta t$. ΔP^+ include only the sources: 1) with window in the range $(w, 2w - \Delta t/R(t)]$ at time t, and 2) receiving a loss indication in the interval $[t, t+\Delta t)$. Note that the probability of receiving multiple loss indications is $o(\Delta t)$, hence, negligible. Hence,

$$\Delta P^+ = \int_{w}^{2w-\Delta t/R(t)} \lambda(\alpha,t)\, \Delta t\, v(\alpha,t) d\alpha + o(\Delta t)$$
$$\frac{\Delta P^+}{\Delta t} \to \int_{w}^{2w} \lambda(\alpha,t) v(\alpha,t) d\alpha. \tag{22}$$

Since $P(w, t+\Delta t) = P(w,t) + \Delta P^+ - \Delta P^-$, we can find (9):

$$\frac{\partial P}{\partial t}(w,t) = \lim_{\Delta t \to 0} \frac{\Delta P^+ - \Delta P^-}{\Delta t}$$
$$= \int_{w}^{2w} \lambda(\alpha,t) v(\alpha,t) d\alpha - \frac{1}{R(t)} v(w,t).$$

Appendix B
Proof of (13) and (14)—Sources With Fast Recovery Mechanisms

The proof is similar to the previous one. Let $v_O(w,t) = \partial P_O(w,t)/\partial w$ and $v_L(w,t) = \partial P_L(w,t)/\partial w$. Consider the sources of class O moving to class L during the interval $[t, t+\Delta t)$; among these, ΔP^+_{OL} will have a window $\leq 2w$ and will contribute to increase $P_L(w,t)$. Analogously to (22):

$$\Delta P^+_{OL} = \int_{1}^{2w} \lambda(\alpha,t)\, \Delta t\, v_O(\alpha,t) d\alpha.$$

The number of sources of class L exceeding w by time $t+\Delta t$ is, analogously to (21):

$$\Delta P^-_L = \int_{w-\Delta t/R(t)}^{w} (1-\lambda(\alpha,t)\Delta t) v_L(\alpha,t) d\alpha. \tag{23}$$

Now consider the population of sources which will leave class L because an RTT is elapsed. We assume an exponential distribution of the departure time of each source from class L, with average $R(t)$. Hence, the number of sources moving from class L to class O will be $\Delta P_{LO} = P_L(w,t)\Delta t/R(t)$, by observing that the fraction of sources that already left class L by the end of Δt is $1 - e^{\Delta t/R(t)} = \Delta t/R(t) + o(\Delta t)$. Now observe that the ΔP^-_{LO}, defined as the number of sources moving from class L to class O and exceeding window w, will include sources counted in both ΔP_{LO} and ΔP^-_L. These source can be derived by ΔP^-_L, since $\Delta P^-_{LO} = \Delta P^-_L \Delta t/R(t)$. Now we are able to add all the contributions:

$$\frac{\partial P_L}{\partial t}(w,t) = \lim_{\Delta t \to 0} \frac{\Delta P^+_{OL} - \Delta P^-_L - \Delta P_{LO} + \Delta P^-_{LO}}{\Delta t}. \tag{24}$$

By recalling (23), we can compute

$$\frac{1}{\Delta t}\left(\Delta P^-_L - \Delta P^-_{LO}\right) = \Delta P^-_L \left(\frac{1}{\Delta t} - \frac{1}{R(t)}\right)$$
$$= \left(\frac{1}{\Delta t} - \frac{1}{R(t)}\right) \int_{w-\Delta t/R(t)}^{w} (1-\lambda(\alpha,t)\Delta t) v_L(\alpha,t) d\alpha$$

whose limit is

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_{w-\Delta t/R(t)}^{w} v_L(\alpha,t) d\alpha = \frac{1}{R(t)} \frac{\partial P_L}{\partial w}(w,t).$$

In other words, ΔP^-_{LO} is negligible with respect to ΔP^-_L. Hence, from (24) we find (14):

$$\frac{\partial P_L}{\partial t}(w,t) = \int_{1}^{2w} \lambda(\alpha,t) v_O(\alpha,t) d\alpha$$
$$- \frac{1}{R(t)} P_L(w,t) - \frac{1}{R(t)} \frac{\partial P_L}{\partial w}(w,t).$$

We can now estimate

$$\frac{\partial P_O}{\partial t}(w,t) = \lim_{\Delta t \to 0} \frac{\Delta P_{LO} - \Delta P^-_{LO} - \Delta P^-_O - \Delta P_{OL}}{\Delta t}$$

where ΔP^-_O are the sources in class O exceeding window w by the time interval Δt and ΔP_{OL} the sources moving from class O to class L, due to losses. It holds that

$$\lim_{\Delta t \to 0} \frac{\Delta P_{OL}}{\Delta t} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_{0}^{w} \lambda(w,t)\, \Delta t\, v_O(\alpha,t) d\alpha$$
$$= \int_{0}^{w} \lambda(\alpha,t) \frac{\partial P_O}{\partial \alpha}(\alpha,t) d\alpha.$$

Analogously to ΔP^-_L, $\Delta P^-_O = (\partial P_O(w,t)/\partial w)/R(t)$. It can be shown that ΔP^-_{LO} is negligible with respect to ΔP_{LO}. Hence, we can obtain (13):

$$\frac{\partial P_O}{\partial t}(w,t) = -\int_{0}^{w} \lambda(\alpha,t) \frac{\partial P_O}{\partial \alpha}(\alpha,t) d\alpha$$
$$- \frac{1}{R(T)} \frac{\partial P_O}{\partial w}(w,t) + \frac{1}{R(T)} P_L(w,t).$$

APPENDIX C
PROOFS OF (16) AND (17)—SOURCES WITH FINITE FLOWS

The only terms which need a formal proof are the ones which model the workload evolution. ΔP is the number of sources which enter $P(w,t,l)$ during a time interval of size Δt because their workload has just decreased. ΔP is given by all the sources with window between 1 and w, and residual workload between l and $l + w\Delta t/R(t)$, being $w/\Delta t$ the instantaneous emission rate of sources with window w. Formally

$$\begin{aligned}\Delta P &= \int_{\alpha=1}^{w}\int_{\beta=l}^{l+w\Delta t/R(t)} \frac{\partial^2 P}{\partial\alpha\partial\beta}(\alpha,t,\beta)d\alpha d\beta \\ &= \int_{\beta=l}^{l+w\Delta t/R(t)} \frac{\partial P}{\partial\beta}(w,t,\beta)d\beta \\ &= P\left(w,t,l+\frac{w}{R(t)}\Delta t\right) - P(w,t,l).\end{aligned}$$

Finally

$$\lim_{\Delta t\to 0}\frac{\Delta P}{\Delta t} = \frac{w}{R(t)}\frac{\partial P}{\partial l}(w,t,l).$$

To account for the sources which stop their activity during the time interval of size Δt, it is enough to set $l = 0$.

APPENDIX D
PROOFS OF (18) AND (19)—SOURCES WITH FINITE FLOWS EXPONENTIALLY DISTRIBUTED

Regarding (18) and (19), we prove formally only those terms accounting the variation of the population $P(w,t)$ due to the variation of the sources residual workload. Consider a time interval of size Δt and a source which does not experience any loss with window w. The probability that this source stops within the interval, i.e., its residual life time is less than Δt, is equal to $1 - \exp\{-\Delta t w/(LR(t))\} \approx \Delta t w/(LR(t))$, thanks to the memoryless property. Then, the contribution of the sources stopping is

$$\lim_{\Delta t\to 0}\frac{\Delta P}{\Delta t} = \int_{\alpha=1}^{+\infty}\frac{\alpha}{LR(t)}\frac{\partial P}{\partial\alpha}(\alpha,t)d\alpha.$$

The final contribution is given by multiplying the previous formula for $(1 - \bar{p}_L(t))$, corresponding to the number of sources not experiencing any losses (those experiencing losses have already been considered in other terms).

APPENDIX E
NUMERICAL SOLUTIONS OF FLUID MODELS

The solutions of our fluid models are obtained by solving numerically the set of differential equations which define the fluid model. Two different approaches have been employed to solve the set of ordinary differential equations (ODEs) describing network queues dynamics and the set of partial differential equations (PDEs) representing the source window dynamics.

Similarly to [5], we solve numerically the ODEs of our model using a fourth-order Runge–Kutta methodology. The Runge–Kutta algorithm is a widely used method to solve ODEs. To solve the source dynamics PDEs of our model, we used a finite differences methodology. Consider a general integro-differential equation of the form

$$\frac{\partial f(w,t)}{\partial t} = K(w,t)\frac{\partial f(w,t)}{\partial w} + \int\phi(w,t)dw \qquad (25)$$

where $\phi(w,t)$ and $K(w,t)$ are continuous function on $\mathbb{R}^2$ (for example in (9), $\phi(w,t) = \lambda(w,t)(\partial f(w,t))/\partial w$ and $K(w,t) = 1/R(t)$).

First we sample $f(w,t)$, $\phi(w,t)$ and $K(w,t)$ onto a bidimensional discrete lattice, defining $f_j^n = f(j\Delta w, nh)$, $\phi_j^n = \phi(j\Delta w, nh)$ and $K_j^n = K(j\Delta w, nh)$; then we approximate the partial derivative:

$$\left.\frac{\partial f(w,t)}{\partial t}\right|_{w=j\Delta w,t=nh} \approx \frac{\partial f_j^n}{\partial t} = \frac{f_j^{n+1} - f_j^{n-1}}{2h}.$$

Similarly, we approximate

$$\left.\frac{\partial f(w,t)}{\partial w}\right|_{w=j\Delta w,t=nh} \approx \frac{\partial f_j^n}{\partial w} = \frac{f_{j+1}^n - f_{j-1}^n}{2\Delta w}.$$

At last, we approximate

$$\int\phi(w,t)dw \approx \sum_j \phi_j^n\Delta w + \sum_j\frac{1}{2}\frac{\partial\phi_j^n}{\partial w}(\Delta w)^2$$

with $\partial\phi_j^n/\partial w$ being the numerical approximation of $\partial\phi(w,t)/\partial w|_{w=j\Delta w,t=nh}$.

In conclusion, we obtain for the PDE the numerical recursion

$$\frac{f_j^{n+1}}{2h} = \frac{f_j^{n-1}}{2h} + K_j^n\frac{\partial f_j^n}{\partial w} + \sum_j\left[\phi_j^n\Delta w + \frac{1}{2}\frac{\partial\phi_j^n}{\partial w}(\Delta w)^2\right].$$

The whole set of differential equations, which defines the fluid model for the considered network, is solved according to the following procedure. At each time iteration h, the parameters $R(t = nh)$ and $\lambda(w, t = nh)$ are evaluated. Note that at time step $t = 0$ we assume that all queues are empty, thus there are no losses and the RTTs account only for fixed propagation delays. Then the equations of the sources' dynamics (PDE) are solved, obtaining as output the amount of fluid that sources inject into the network. Next, the network queues' dynamics (ODE) are solved using as input the amount of fluid injected by sources. The state variables of the queues are then used to update the RTTs and loss rates perceived by the sources. We notice that the proposed scheme is very similar to the one used in [5].

REFERENCES

[1] F. Baccelli and D. Hong, "Interaction of TCP flows as billiards," in *Proc. IEEE INFOCOM*, San Francisco, CA, 2003, pp. 895–905.
[2] ——, "Flow level simulation of large IP networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, 2003, pp. 1911–1921.
[3] S. Misra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 151–160.
[4] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. IEEE INFOCOM*, Anchorage, AK, 2001, pp. 1726–1734.

[5] Y. Liu, F. Lo Presti, V. Misra, and D. Towsley, "Fluid models and solutions for large-scale IP networks," in *Proc. ACM SIGMETRICS*, San Diego, CA, Jun. 2003, pp. 90–101.
[6] S. Deb, S. Shakkottai, and R. Srikant, "Stability and convergence of TCP-like congestion controllers in a many-flows regime," in *Proc. IEEE INFOCOM*, San Francisco, CA, 2003, pp. 884–894.
[7] S. Shakkottai and R. Srikant, "How good are deterministic fluid models of internet congestion control?," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 497–505.
[8] P. Tinnakornsrisuphap and A. Makowski, "Limit behavior of ECN/RED gateways under a large number of TCP flows," in *Proc. IEEE INFOCOM*, San Francisco, CA, 2003, pp. 873–883.
[9] F. Baccelli, D. R. McDonald, and J. Reynier, "A mean-field model for multiple TCP connections through a buffer implementing RED," *Perform. Eval.*, vol. 49, no. 1/4, pp. 77–97, 2002.
[10] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
[11] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Vancouver, Canada, Sep. 1988, pp. 314–329.
[12] L. Zhang and D. Clark, "Oscillating behavior of network traffic: A Case study simulation," *Internetworking: Research and Experience*, vol. 1, no. 2, pp. 101–112, 1990.
[13] M. Ajmone Marsan, M. Garetto, P. Giaccone, E. Leonardi, E. Schiattarella, and A. Tarello. Using partial differential equations to model TCP mice and elephants in large IP network. [Online]. Available: http://www.tlc-networks.polito.it/database/ricer.htm
[14] A. Feldmann and W. Whitt, "Fitting mixtures of exponentials to long-tail distributions to analyze network performance models," in *Proc. IEEE INFOCOM*, Kobe, Japan, 1997, pp. 1096–1104.
[15] M. Garetto and D. Towsley, "Modeling, simulation and measurements of queueing delay under long-tail internet traffic," in *Proc. ACM SIGMETRICS*, San Diego, CA, Jun. 2003, pp. 47–57.
[16] S. B. Fredj, T. Bonald, A. Proutiere, G. Régnié, and J. Roberts, "Statistical bandwidth sharing: A study of congestion at flow level," in *Proc. ACM SIGCOMM*, San Diego, CA, 2001, pp. 111–122.

Marco Ajmone Marsan (F'99) holds degrees in electronic engineering from the Politecnico di Torino, Torino, Italy, and the University of California at Los Angeles (UCLA). In 2002, he was awarded an *Honoris Causa* degree in telecommunication networks from the Budapest University of Technology and Economics, Budapest, Hungary.

He is a Full Professor at the Electronics Department of Politecnico di Torino, in Italy. Since September 2002, he has been the Director of the Institute for Electronics, Information Engineering and Telecommunications of the National Research Council. From November 1975 to October 1987, he was at the Electronics Department of Politecnico di Torino, first as a Researcher, then as an Associate Professor. From November 1987 to October 1990, he was a Full Professor at the Computer Science Department of the University of Milan, Italy. During the summers of 1980 and 1981, he was with the Research in Distributed Processing Group, Computer Science Department, UCLA. During summer 1998, he was an Erskine Fellow at the Computer Science Department of the University of Canterbury, New Zealand. He has coauthored over 300 journal and conference papers in the areas of telecommunications and computer science, as well as the two books *Performance Models of Multiprocessor Systems* (MIT Press), and *Modeling with Generalized Stochastic Petri Nets* (Wiley). He has been the principal investigator in national and international research projects in the field of telecommunication networks. His current interests are in the fields of performance evaluation of communication networks and their protocols.

Dr. Ajmone Marsan received the Best Paper Award at the Third International Conference on Distributed Computing Systems, Miami, FL, in 1982. 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 *Computer Networks*.

Michele Garetto (S'01–M'04) received the Dr.Ing. degree in Telecommunication Engineering and the Ph.D. degree in electronic and telecommunication engineering, both from Politecnico di Torino, Italy, in 2000 and 2004, respectively.

In 2002, he was a visiting scholar with the Networks group of the University of Massachusetts, Amherst, and in 2004 he held a Postdoctoral position at Rice University, Houston, TX. His research interests are in the field of performance evaluation of wired and wireless communication networks.

Paolo Giaccone (S'00–M'02) received the Dr.Ing. and Ph.D. degrees in telecommunications engineering from Politecnico di Torino, Italy, in 1998 and 2001, respectively.

He is Assistant Professor in the Electronics Department, Politecnico di Torino. During the summer 1998, he visited the High Speed Networks Research Group at Lucent Technology, Holmdel, NJ. During 2000–2001 and during summer 2002, he visited the Electrical Engineering Department, Stanford University. he held a Postdoctoral position at Politecnico di Torino between 2001 and 2002, and during summer 2002 at Stanford University. His main area of interest is the design of scheduling policies for high-performance routers.

Emilio Leonardi (S'94–M'99) received the Dr.Ing. degree in electronics engineering and the Ph.D. degree in telecommunications engineering, both from Politecnico di Torino, Italy, in 1991 and 1995, respectively.

He is currently an Associate Professor at the Dipartimento di Elettronica, Politecnico di Torino. In 1995, he visited the Computer Science Department, 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, Burlingame, CA. His research interests are in the fields of performance evaluation of communication networks, switching architectures, and all-optical networks.

Enrico Schiattarella (S'01) received the Dr.Ing. degree in electrical engineering from Politecnico di Torino, Italy, in 2002, where he is currently pursuing the Ph.D. degree.

From October 2001 to June 2002, he visited Cisco Systems, in summer 2003, Alcatel R&I, Marcoussis, France, and from March to November 2004, the Server Interconnect Fabrics group at IBM Zurich Research Lab, Switzerland. His main research interests are in high-performance switching and routing.

Alessandro Tarello (S'03) received the M.Sc. degree in communication engineering from Politecnico di Torino, Torino, Italy, in 2002, where he is currently pursuing the Ph.D. degree.

From January to December 2004, he visited the Laboratory for Information and Decision Systems, MIT, Cambridge, MA.