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

M.Ajmone Marsan, M.Garetto, P.Giaccone, E.Leonardi, E.Schiattarella, A.Tarello
Dipartimento di Elettronica, Politecnico di Torino, Italy
e-mail: {ajmone,garetto,giaccone,leonardi,schiattarella,tarello}@mail.tlc.polito.it

***Abstract*— Fluid models of IP networks have been recently proposed as a way to break the scalability barrier of traditional discrete state-space models, both simulative (e.g., *ns-2*) and analytical (e.g., queues and Markov chains).**

Fluid models adopt an abstract deterministic description of the average network dynamics through a set of ordinary differential equations that are then solved numerically, obtaining estimates of the time-dependent network behavior. However, an important limit of the fluid model approaches presented so far in the literature is their unnatural representation of scenarios comprising the short-lived TCP flows that dominate in today's Internet.

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 little 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* simulations proves the accuracy of the proposed modeling approach.

I. Introduction

Traditional approaches to performance evaluation of telecommunication networks in general, and of packet networks in particular (we specifically refer to IP networks in this paper), have normally relied on attempts to describe as closely as possible the dynamics of network elements over a discrete state-space. We shall refer to these approaches as 'discrete models'. Discrete models are quite a natural choice in light of the fact that the operations of traffic sources, of switches, and of protocols, are normally governed by finite-state machines, whose dynamics determine the IP network performance. However, discrete models, requiring the description of the dynamics of the different network elements over their discrete state-spaces (accounting for the dependencies induced by network control algorithms, by end-to-end protocols, and by the traffic flowing through several network elements), suffer from limited scalability, thus allowing only the performance analysis of rather small networking setups. This is the reason why only toy topologies are normally considered in IP network performance studies, and models almost invariably concentrate on a very limited subset of the network protocol stack.

This work was supported by the Italian Ministry for Education, University and Research, within the TANGO project.

Today, IP networks have become extraordinarily large and complex, and, in order to predict the effects on performance of new topologies or protocols, scalable modeling approaches are a must. To obtain scalability, researchers often model one network element (a protocol layer or a router) with a high degree of accuracy, whereas the remaining network elements are either neglected, or modeled in a very coarse manner. This approach may sometimes produce acceptable results, but many cases exist where the interaction among network elements is a focal point for performance issues (for example, in a satellite IP network, the interaction between TCP and the ARQ protocol at layer 2 is crucial).

The above comments apply to analytical models as well as simulation. Indeed, traditional discrete models for simulation are based on a detailed packet-level description of the network, and consequently suffer of scaling problems resulting from the growth of CPU times and memory requirements beyond the capability of available machines. On the analytical side, the application of discrete-state probabilistic models to the analysis of sizable portions of the Internet appears prohibitive, although such models (continuous-time or discrete-time Markov chains and queueing models in particular) have traditionally been the mathematical tool of choice in the networking field.

Network Calculus [1], [2] is able to cope with full-size network performance evaluation. It allows an abstract, deterministic, flow-level, worst-case analysis of the network dynamics, thus greatly simplifying the study of complex networking setups, but up to now has produced rather loose performance bounds under simplified traffic assumptions.

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. This new approach, that is often called 'fluid models', adopts an abstract deterministic description of the average network dynamics through a set of differential equations [3], [4], [5], [6], [7], thus neglecting the short-term, 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 'ele-

phants'). 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 grows very large [3], [8], [9], [10].

An important limit of the fluid model approaches presented so far in the literature is their unnatural representation of scenarios comprising the short-lived TCP flows (commonly called 'mice') that dominate in today's Internet.

In this paper we develop a new fluid model approach based on *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, 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, by first discussing the simplest version, and then progressively extending it, to cope with finite window sizes, fast recovery, drop-tail buffers, and TCP mice. 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. Finally, Section V concludes the paper.

II. The MGT Fluid Model of IP Networks

In [5], [6], [7], 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 [11]) active queue management (AQM) scheme. Their approach (that we name 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 non-homogeneous 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 AIMD (Additive Increase, Multiplicative Decrease). 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 packet queue in the kth buffer at time t; the temporal evolution of the queue level is described by:

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

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 [5] for RED buffers.

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

$$T_k(t) = Q_k(t)/C$$

If $\mathcal{F}_k$ indicates the set of elephants 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:

$$A_k(t) = \sum_{i \in \mathcal{F}_k} A_k^i(t) \ , \ \int_0^{t+T_k(t)} D_k(a)da = \int_0^t A_k(a)da$$

$$\int_0^{t+T_k(t)} D_k^i(a)da = \int_0^t A_k^i(a)da,$$

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. By differentiating the last equation:

$$D_k^i(t + T_k(t))\left(1 + \frac{dT_k(t)}{dt}\right) = A_k^i(t)$$

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:

$$R_i\left(t + g_i + \sum_{h=1}^{L_i} T_{k(h,i)}(t_{k(h,i)})\right) = g_i + \sum_{h=1}^{L_i} T_{k(h,i)}(t_{k(h,i)}) \tag{2}$$

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 source reaches the hth buffer along its path P_i. We have:

$$t_{k(h,i)} = t_{k(h-1,i)} + T_{k(h-1,i)}(t_{k(h-1,i)}) \tag{3}$$

The loss indicator rate is instead given by:

$$\lambda_i\left(t + R_i(t)\right) = \alpha \frac{W_i(t)}{R_i(t)} p_i^F(t) \tag{4}$$

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:

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

Finally:

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

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 [4]. 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 (4), which is necessary to compensate for the use of the average window size, instead of the window size distribution, must be set according to an empirical process.

III. Previous Work on Fluid Models

Fluid models have been recently proposed [3], [4], [5], [6], [7] as a useful approach to estimate the performance of large IP networks loaded with TCP traffic. In particular, fluid models were proposed as a viable alternative to packet-based simulators, since the complexity of fluid models (i.e., the number of equations to be solved) is independent of the number of TCP flows and of link capacities.

To the best of our knowledge, fluid models were first proposed in [5] to study the interaction between TCP elephants and a RED buffer in a packet network consisting of just one bottleneck link. In [7] the authors have recently extended their model to consider general multi-bottleneck topologies comprising RED routers.

As we have already observed, the equations reported in Section II briefly summarize the fluid model proposed in [7], which constitute the starting point for our work. This set of ordinary differential equations must be solved numerically, using standard discretization techniques.

In [3], [4] an alternative fluid model is 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 the number of TCP flows. An extension that allows considering TCP mice is also proposed in [3], [4]. 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.

IV. Modeling a large population of TCP sources

The class of fluid models that we propose in this paper differs from previous proposals 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: (i) 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, (ii) 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 try to 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.

[1]Equation (2) 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 $t_{k(h,i)}$.

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 described by the following equation, for $w \geq 1$:

$$\frac{\partial P(w,t)}{\partial t} = \int_w^{2w} \lambda(\alpha,t)\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha - \frac{1}{R(t)}\frac{\partial P(w,t)}{\partial w} \quad (5)$$

where $\lambda(w,t)$ is the loss indication rate. A formal derivation of (5) is given in Appendix A. 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: (i) the integral accounts for the growth rate of $P(w,t)$ due to sources with window between w and $2w$ that experience losses; (ii) 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 (4):

$$\lambda(w,t) = \frac{wp^F(t)}{R(t)} \quad (6)$$

in which the current window of the sources that emitted the lost fluid approximates the window 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 TCP windows impact on network stationary behavior: the MGT model only evaluates the mean value of TCP windows, while this model evaluates their distribution.

B. Accounting for the maximum window size

We now extend the basic model of (5) to account for the maximum window size of TCP sources, that we denote by W^{max}. It holds:

$$\frac{\partial P(w,t)}{\partial t} = \int_w^{\min(2w,W^{max})} \lambda(\alpha,t)\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha + \\ + \lambda(W^{max},t)P_{max}(t)u(w-W^{max}/2) - \frac{1}{R(t)}\frac{\partial P(w,t)}{\partial w} \quad (7)$$

for $1 \leq w < W^{max}$, 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}.

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

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

$$\frac{dP_{max}(t)}{dt} = \\ = \frac{1}{R(t)} \lim_{w \uparrow W^{max}} \frac{\partial P(w,t)}{\partial w} - \lambda(W^{max},t)P_{max}(t) \quad (8)$$

with the boundary conditions: $P(1^-,t) = 0$ and $\lim_{w\uparrow W^{max}} P(w,t) + P_{max}(t) = N$. The derivation of (7) is very similar to that of (5). The first term in (7) 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 (7) 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

In this subsection we discuss some numerical results referring to the mathematical model in (7). Before proceeding we notice that all the results shown in this paper were obtained by solving numerically the model. For this purpose we applied standard discretization techniques; in particular, a first-order finite differences method for the sources equations and a second-order Runge-Kutta method for the queue equations.

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 8 TCP elephants saturating the link capacity ($C = 100$ Mbps), 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 10000 bits. In Fig. 1 we show the window size probability density function of elephants in class 2 predicted by our model. In Table I we compare the average window size, the average queue length and the loss probability for the model and the *ns* 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 [5], [6], [7], 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 results of *ns* simulations for the same setup, reported in Table I and in Fig. 2 for comparison, clearly show that the fluid model is quite accurate.

D. Considering Fast Recovery

Newer versions of TCP (such as NewReno - see RFCs 2001, 2581, 2582) avoid halving the window more than once for

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

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.

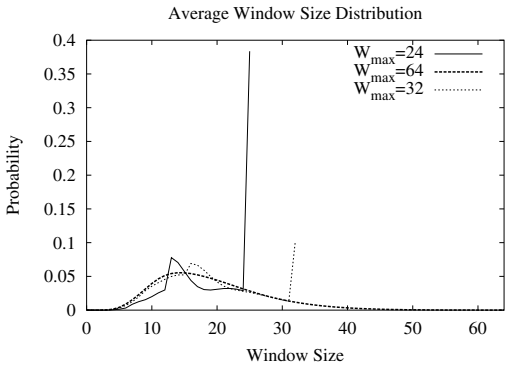

Fig. 1. Fluid model: average window size distribution for 8 TCP elephants traversing a single bottleneck link with RED buffer, varying their maximum window size; these TCP flows compete with 8 other TCP elephants with maximum window size 64.

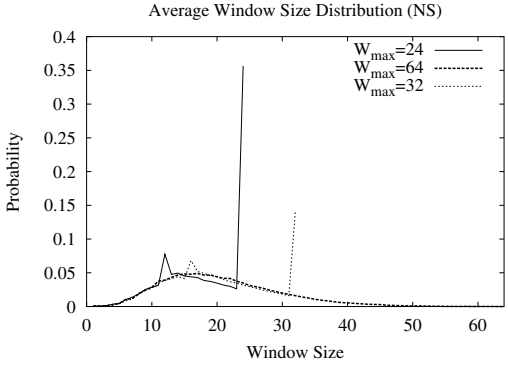

Fig. 2. ns simulator: same as Fig. 1

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

We can write:

$$\frac{\partial P_O(w,t)}{\partial t} = -\int_1^w \lambda(\alpha,t)\frac{\partial P_O(\alpha,t)}{\partial \alpha}d\alpha - \frac{1}{R(t)}\frac{\partial P_O(w,t)}{\partial w} + \frac{1}{R(t)}P_L(w,t) \quad (9)$$

[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.

$$\frac{\partial P_L(w,t)}{\partial t} = +\int_1^{2w} \lambda(\alpha,t)\frac{\partial P_O(\alpha,t)}{\partial \alpha}d\alpha - \frac{1}{R(t)}P_L(w,t) - \frac{1}{R(t)}\frac{\partial P_L(w,t)}{\partial w} \quad (10)$$

A formal derivation of (9) and (10) is reported in Appendix B. An intuitive explanation of the two equations can be provided as follows. In the right hand side of (9), 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 (10), 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: (i) sources moving to class O from class L after a RTT without losses, (ii) 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 Appendix E.

E. Modeling drop-tail buffers

As we have already mentioned, a fluid model for the description of RED AQM schemes was originally proposed in [5]. 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.

The level of detail in the description of the TCP sources dynamics adopted in this paper allows an easy description of

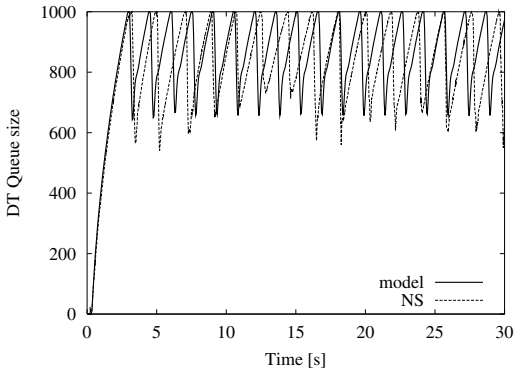

Fig. 3. Fluid model and ns simulator: queue size evolution for one bottleneck link with data rate $C = 100$ Mbps, propagation delay 30 ms, fed by a drop-tail buffer with capacity equal to 1000 packets, and traversed by 30 TCP elephants, with maximum window size 64 packets.

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

TABLE II

AVERAGE WINDOW SIZE (AWS), AVERAGE QUEUE LENGTH (AQL) AND AVERAGE LOSS PROBABILITY (ALP) FOR THE SAME SETUP OF FIG. 3

the time-dependent behavior of the packet loss probability:

$$p_k(t) = \frac{\max(0, A_k(t) - C)}{A_k(t)} \mathbb{1}_{\{Q_k(t)=B_k\}} \qquad (11)$$

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 $\mathbb{1}_{\{\cdot\}}$ the indicator function.

A different approach is used in [3] and [4] 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. This 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 $C = 100$ Mbps, 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. 3 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 [12], [13].

The results of *ns* simulations are reported in Fig. 3 and Table II for comparison, and again show that the fluid model is accurate.

The second scenario we consider is a network topology comprising two links, the first fed by a RED buffer, the second

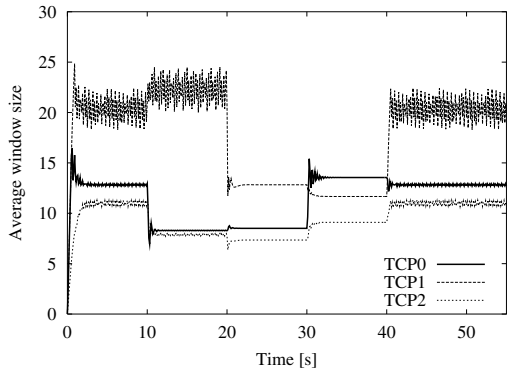

Fig. 4. Fluid model: window size evolution for three long-lived TCP flows with interfering UDP traffic.

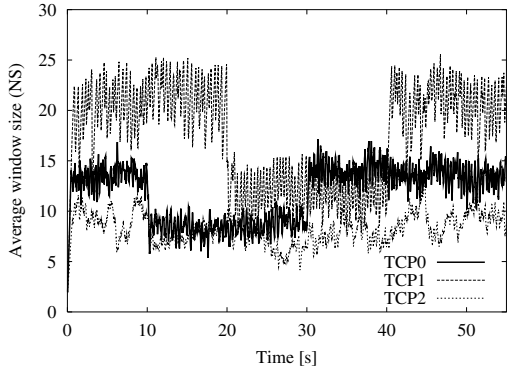

Fig. 5. ns simulator: same as Fig. 4

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. 4 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* simulations for the same setup are reported in Fig. 5 for comparison, and once more show that the fluid model is quite accurate (in addition, in Fig. 6 and 7 we overlap the curves of the model and the *ns* simulator for the TCP0 and TCP1 elephants).

These results prove that our model can cope with both controlled (TCP) and uncontrolled (UDP) long-lived flows,

[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.

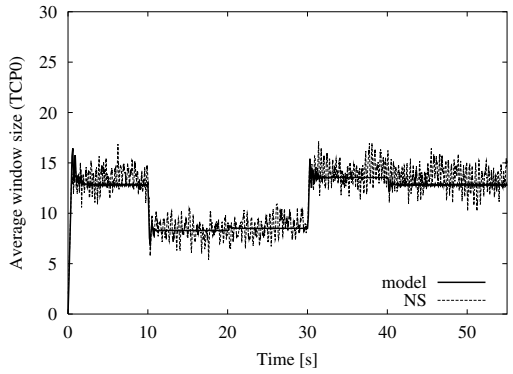

Fig. 6. Overlap of TCP0 curves from Fig. 4 and 5.

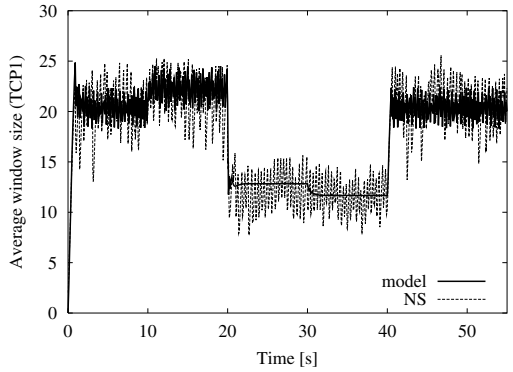

Fig. 7. Overlap of TCP1 curves from Fig. 4 and 5.

and is capable of predicting the TCP transient effects due to the presence of on-off interfering sources.

G. 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 at the start of the TCP connection. 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.

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:

$$\frac{\partial P(w,t,l)}{\partial t} = -\frac{1}{R(t)}\frac{\partial P(w,t,l)}{\partial w} - \frac{w}{R(t)}\frac{\partial P(w,t,l)}{\partial l}\bigg|_{l=0} + \frac{w}{R(t)}\frac{\partial P(w,t,l)}{\partial l} + \int_w^{2w}\lambda(\alpha,t)\frac{\partial P(\alpha,t,l-1)}{\partial \alpha}d\alpha + \int_1^{2w}\lambda(\alpha,t)\frac{\partial P_s(\alpha,t,l-1)}{\partial \alpha}d\alpha \quad (12)$$

$$\frac{\partial P_s(w,t,l)}{\partial t} = -\frac{w}{R(t)}\frac{\partial P_s(w,t,l)}{\partial w} - \frac{w}{R(t)}\frac{\partial P_s(w,t,l)}{\partial l}\bigg|_{l=0} + \frac{w}{R(t)}\frac{\partial P_s(w,t,l)}{\partial l} - \int_1^{w}\lambda(\alpha,t)\frac{\partial P_s(\alpha,t,l-1)}{\partial \alpha}d\alpha + \gamma(t,l) \quad (13)$$

A formal proof of these equations is given in Appendix C. An intuitive explanation is as follows. In (12), the first two terms on the right hand side account for the decrease rate of $P(w,t,l)$ due to: (i) sources increasing their rate (first term), (ii) 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 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 (13) is very similar to (12), 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 (12)), and 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 size, thus providing the correct level of detail for the analysis of this type of TCP flows. Indeed, TCP mice start in slow-start, with window 1, and then their window evolves according to (12) and (13).

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:

$$\frac{\partial P(w,t)}{\partial t} = -\frac{1}{R(t)}\frac{\partial P(w,t)}{\partial w} - \frac{(1-\bar{p}_L(t))}{R(t)L}\int_1^w \alpha\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha + \int_w^{2w}\lambda(\alpha,t)\frac{\partial P(\alpha,t)}{\partial \alpha}d\alpha + \int_1^{2w}\lambda(\alpha,t)\frac{\partial P_s(\alpha,t)}{\partial \alpha}d\alpha \quad (14)$$

$$\frac{\partial P_s(w,t)}{\partial t} = -\frac{w}{R(t)}\frac{\partial P_s(w,t)}{\partial w} - \frac{(1-\bar{p}_L(t))}{R(t)L}\int_1^w \alpha \frac{\partial P_s(\alpha,t)}{\partial \alpha} d\alpha - \int_1^w \lambda(\alpha,t)\frac{\partial P_s(\alpha,t)}{\partial \alpha} d\alpha + \gamma(t) \quad (15)$$

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 [5], [6] to evaluate the average loss probability in a RED queue; we obtain:

$$\frac{\partial \bar{p}_L(t)}{\partial t} = -\frac{\bar{w}(t)}{LR(t)}\bar{p}_L(t) + \frac{\bar{w}(t)}{LR(t)}p^F(t) \quad (16)$$

being $p^F(t)$ the instantaneous loss probability, defined in (4), and $\bar{w}(t)$ the average window size at time t.

We wish to stress the fact that (14)-(16) provide quite a powerful tool for an efficient representation of TCP mice, since a wide range of distributions (including those incorporating long range dependence) can be approximated with a good degree of accuracy by a mixture of exponential distributions [14].

H. Randomness in fluid models

As we have observed, fluid models provide a deterministic, phenomenological description of the network behavior, thus departing from the common approaches of attempting a probabilistic description of the network dynamics over a huge state space. However, in networking scenarios with only TCP mice, the pure determinism of fluid models fails to provide useful information about the buffering phenomena within the network: if buffers are underloaded, their corresponding fluid model is constantly empty, whereas if buffers are overloaded, their corresponding fluid model grows to infinity.

That is, the intrinsic determinism of fluid models does not allow them to grasp the random fluctuations in the mice arrival pattern and size which cause buffer occupancies to grow above zero without diverging.

In order to overcome this limit of fluid models in the analysis of the performance of TCP mice, we believe that some randomness has to be introduced within fluid models. This can be done at several levels.

- The deterministic mice arrival rate $\gamma(t)$ in (15) can be replaced by a Poisson counter with average $\gamma(t)$, thus making (15) a stochastic partial differential equation.
- The deterministic completion process of TCP connections can be replaced by an inhomogeneous Poisson process whose average at time t is represented by the second term of (14). Note that this term suggests that, due to retransmissions, the completion time of connections increases as their loss probability grows.
- Instead of assuming the workload emitted by TCP sources to be a continuous deterministic fluid process with rate $W_i(t)/R_i(t)$, it can be taken to be a Poisson point process (possibly with batch arrivals) with the same rate.

	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	100	508	45.8	806	0.0027
	200	512	34.9	806	0.0054
	400	750	12.9	926	0.024

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 SUBSECTION IV-I.

Of course, this is only a preliminary attempt to introduce randomness in fluid models, so as to be able to study 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 introducing randomness within fluid models without losing the property of independence of complexity with respect to the number of flows.

I. Experiments with mice

In this subsection we discuss results for networking scenarios comprising TCP mice. First, we consider a case in which both mice and elephants coexist. 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 $C = 100$ Mbps, the propagation delay between the TCP sources and the buffer is 30 ms. 20 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/s. The queue size evolution is similar to that already presented in Fig. 3. The presence of elephants is crucial in order to saturate the link bandwidth, because they consume the capacity that is not used by mice, which are practically uncontrolled. Indeed, in 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 flows, obtained from the average number of active mice with Little's theorem.

Table III and Fig. 8 also report the results of *ns* simulations for the same setup, for comparison: the fluid model is quite accurate in this case too.

If infinite flows are removed from the scenario, as we discussed in the previous subsection, fluid models cannot provide useful information about the network performance. Consider a single bottleneck link fed by a drop-tail buffer, with capacity equal to 256 packets. The link capacity C is 100 Mbps, while the propagation delay between sources and buffer is 30 ms; there are 3 classes of TCP mice: they all have geometrically distributed size, 89% with average 10 packets, 10% with average 100, and 1% with average 1000. The maximum window size is set to 64 packets for all TCP sources. Experiments with loads equal to 0.6, 0.8 and 0.95 show that the buffer is always empty, while loads over 1.0

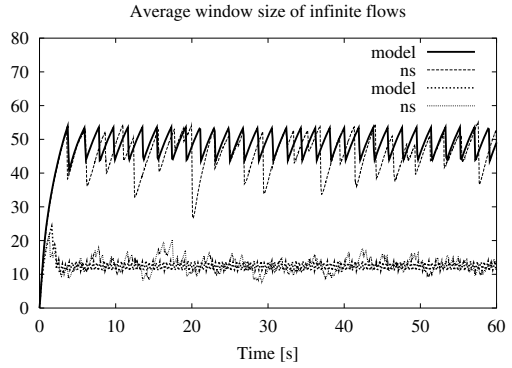

Fig. 8. Average window size evolution for elephants competing with mice on a single bottleneck link; the average window size decreases when the mice arrival rate grows from 100 to 400 connections/s.

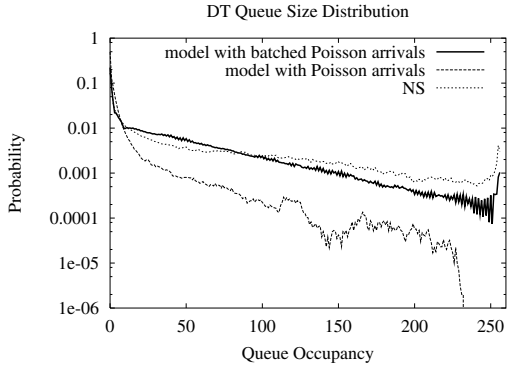

Fig. 9. Queue size distribution for single drop tail bottleneck, varying the random process modeling the workload emitted by the TCP sources; comparison with ns simulator.

	ALP	AQL	ACT
Poisson	0	13.6	118, 213, 652
batched Poisson	0.0032	39.7	126, 233, 892
ns	0.003	54.2	136, 274, 1097

TABLE IV

AVERAGE LOSS PROBABILITY (ALP), AVERAGE QUEUE LENGTH (AQL) AND AVERAGE COMPLETION TIMES (ACT) OF THE THREE CLASSES OF MICE FOR THE SETUP OF SUBSECTION IV-I, HAVING INTRODUCED RANDOM ELEMENTS.

obviously saturate the buffer (plots are not reported here).

If we introduce randomness in the fluid model, using the approach described in the previous subsection, we obtain the results shown in Fig. 9, which refer to load 0.8: the buffer occupancy distribution still has a peak in 0, but the variance has increased.

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 do not match well the results obtained with the *ns* simulator, as can be observed in Table IV and Fig. 9. However, the performance predictions obtained with the fluid model become extremely accurate when the workload emitted by TCP sources is taken to be a Poisson process with batch arrivals, with batch size distribution derived from the window size distribution of TCP mice. This approach derives from recent results about the close relationship existing between the burstiness of the traffic generated by mice and their window size [15].

J. Experiments with a real topology

In order to conclude our validation of the fluid models proposed in this paper, we present some results referring to a mesh network topology (depicted in Fig. 10), that mimics the Italian Academic and Research Network (named GARR). The network backbone comprises 4 core routers connected by 5 links with rates equal to either 1 Gbps or 2.5 Gbps, and propagation delay 30 ms. Each core router is connected to edge routers through 622 Mbps links, whose propagation delays are comprised between 5 and 25 ms. The total number of edge routers is 18. In addition, a 2.5 Gbps 100 ms transoceanic link is connected to one of the four core routers.

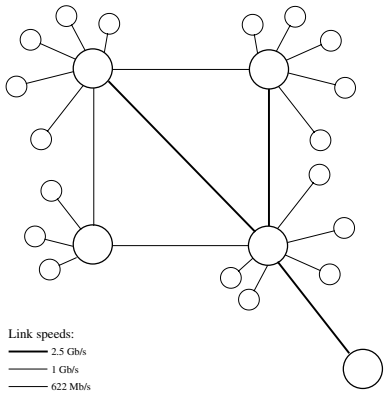

Fig. 10. simulated GARR network topology

255 elephants enter the network through edge routers, and 1350 elephants through the transoceanic link. In addition, 850 mice/s enter through edge routers, and 5400 mice/s through the transoceanic link. The mice length is taken to be geometrically distributed with average equal to 20 packets. All the mice and elephants are uniformly distributed among all the possible sources and destinations, generating about 350 classes of flows (for either mice or elephants).

The numerical solution of the network dynamics over a period of one minute implies the investigation of the dynamics of about 1.2 millions TCP flows, and requires just few minutes over a standard 1 GHz PC, thanks to the excellent scalability properties of the model.

Figs. 11 and 12 report the throughput and completion time performance for TCP elephants and mice, respectively, in decreasing order. Both cases of a deterministic fluid model and of a model with randomness are considered. In this particular case, the presence of elephants makes the impact of randomness marginal. However, if we remove the elephants, as we already saw, the randomness becomes necessary in order to observe interesting phenomena.

V. CONCLUSIONS

In this paper we have proposed a new fluid model approach for the investigation of the performance of IP networks loaded

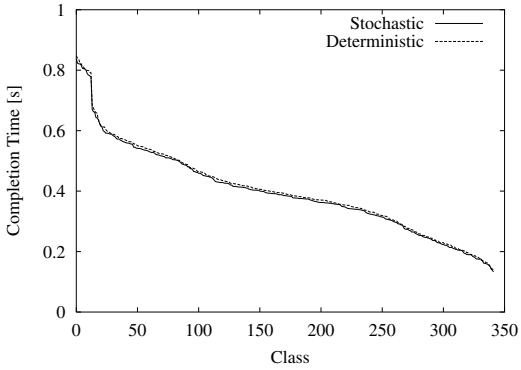

Fig. 11. Average completion time per class, in decreasing order, for mice present in the GARR topology.

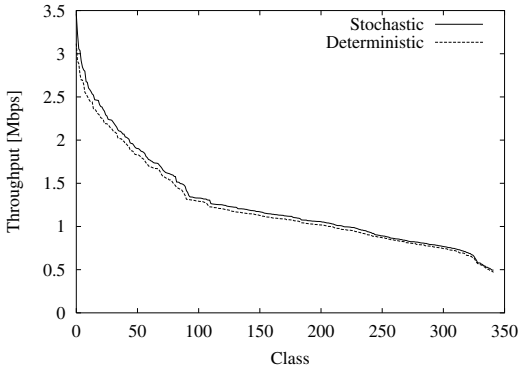

Fig. 12. Average throughput per class, in decreasing order, for elephants present in the GARR topology

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* simulations in the cases in which the latter are feasible, proving both the accuracy and the scalability of the proposed modeling approach. In addition, we applied the proposed fluid modeling approach to a realistic network, showing that the solution of reasonable size networks can be obtained with limited computational complexity.

Further work on this topic will include the investigation of larger networking setups, the optimization of the numerical solution techniques, as well as a more detailed investigation of the possible approaches to introduce randomness in the modeling paradigm, so as to improve the accuracy in the description of networks where only mice are present.

REFERENCES

[1] R.Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation" *IEEE/ACM Transaction on Information Theory*, vol. 37, n. 1, pp. 114-131, January 1991.

[2] J.Y Le Boudec, P.Thiran, "Network Calculus: a Theory of Deterministic Queuing Systems for the Internet", Springer Verlag LNCS 2050, June 2001.

[3] F.Baccelli, D.Hong, "Interaction of TCP Flows as Billiards", *IEEE Infocom 2003,* San Francisco, CA, March 2003.

[4] F.Baccelli, D.Hong, "Flow Level Simulation of Large IP Networks", *IEEE Infocom 2003,* San Francisco, CA, March 2003.

[5] S.Misra, W.B.Gong, D. Towsley, "Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", *ACM SIGCOMM 2000,* Stockholm, Sweden, August 2000.

[6] C.V.Hollot, V.Misra, D.Towsley, W.B.Gong, "On the Design of Improved Controllers for AQM Routers Supporting TCP Flows", *IEEE Infocom 2001,* Anchorage, Alaska, USA, April 2001.

[7] Y.Liu, F.Lo Presti, V.Misra, D.Towsley, "Fluid Models and Solutions for Large-Scale IP Networks", *ACM Sigmetrics 2003,* San Diego, CA, June 2003.

[8] S.Deb, S.Shakkottai, R.Srikant, "Stability and Convergence of TCP-like Congestion Controllers in a Many-Flows Regime", *IEEE Infocom 2003,* San Francisco, CA, March 2003.

[9] S.Shakkottai, R.Srikant, "How Good are Deterministic Fluid Models of Internet Congestion Control?" *IEEE INFOCOM 2002,* New York, June 2002.

[10] P.Tinnakornsrisuphap, A.Makowski, "Limit Behavior of ECN/RED Gateways Under a Large Number of TCP Flows", *IEEE Infocom 2003,* San Francisco, CA, March 2003.

[11] S.Floyd, V.Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, n. 4, pp. 397-413, August 1993.

[12] V.Jacobson, "Congestion Avoidance and Control" *ACM Sigcomm 1998,* Vancouver, Canada, September 1998.

[13] L.Zhang, D.Clark, "Oscillating Behavior of Network Traffic: a Case Study Simulation", Internetworking: Research and Experience, vol. 1, n. 2, pp. 101-112, 1990.

[14] A. Feldmann, W. Whitt, " Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models", *IEEE Infocom 97*, Kobe, Japan, April 1997.

[15] M.Garetto, D.Towsley, "Modeling, Simulation and Measurements of Queuing Delay Under Long-Tail Internet Traffic", *ACM Sigmetrics 2003,* San Diego, CA, June 2003.

[16] J.Padhye, V.Firoiu, D.Towsley, J.Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", *ACM Sigcomm 1998,* Vancouver, Canada, September 1998.

APPENDIX

A. Proof of Eq. (5) - 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 $\geq w - \Delta t/R(t)$, since they will exceed w by time $t+\Delta t$. If we assume to model (locally) the loss indication process 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:

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

$$\frac{\Delta P^-}{\Delta t} \rightarrow \frac{1}{R(t)} v(w,t) \qquad (17)$$

Let now 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 (i) with window in the range $(w, 2w - \Delta t/R(t)]$ at time t, and (ii) 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,

$$\Delta P^+ = \int_w^{2w-\Delta t/R(t)} \lambda(\alpha,t)\ \Delta t\ v(\alpha,t)d\alpha + o(\Delta t)$$
$$\frac{\Delta P^+}{\Delta t} \to \int_w^{2w} \lambda(\alpha,t)v(\alpha,t)d\alpha \quad (18)$$

Since $P(w,t+\Delta t) = P(w,t) + \Delta P^+ - \Delta P^-$, we can find (5):

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

B. Proof of Eq.(9) and (10) - 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 (A):

$$\Delta P^+_{OL} = \int_0^{2w} \lambda(\alpha,t)\ \Delta t\ v_O(\alpha,t)d\alpha$$

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

$$\Delta P^-_L = \int_{w-\Delta t/R(t)}^{w} (1-\lambda(\alpha,t)\Delta t)v_L(\alpha,t)d\alpha \quad (19)$$

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 number of sources that already left class L by the end of Δt will be $1 - e^{\Delta t/R(t)} = \Delta t/R(t) + o(\Delta t)$. 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 possible contributions:

$$\frac{\partial P_L}{\partial t}(w,t) = \lim_{\Delta t\to 0}\frac{\Delta P^+_{OL} - \Delta P^-_L - \Delta P_{LO} + \Delta P^-_{LO}}{\Delta t} \quad (20)$$

By recalling (19), 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:

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

In other words, ΔP^-_{LO} is negligible with respect to ΔP^-_L. Hence, from (20) we find (10):

$$\frac{\partial P_L}{\partial t}(w,t) = \int_0^{2w}\lambda(\alpha,t)v_O(\alpha,t)d\alpha \\ - \frac{1}{R(t)}P_L(w,t) - \frac{1}{R(t)}\frac{\partial P_L}{\partial w}(w,t)$$

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:

$$\lim_{\Delta t\to 0}\frac{\Delta P_{OL}}{\Delta t} = \lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_0^w \lambda(w,t)\ \Delta t\ v_O(\alpha,t)d\alpha = \\ = \int_0^w \lambda(\alpha,t)\frac{\partial P_O}{\partial \alpha}(\alpha,t)d\alpha$$

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 Eq. (9):

$$\frac{\partial P_O}{\partial t}(w,t) = -\int_0^w \lambda(\alpha,t)\frac{\partial P_O}{\partial \alpha}(\alpha,t)d\alpha \\ - \frac{1}{R(T)}\frac{\partial P_O}{\partial w}(w,t) + \frac{1}{R(T)}P_L(w,t)$$

C. Proofs of Eq. (12) and (13) - 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,

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

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 $l = 0$.

D. Proofs of Eq. (14) and (15) - sources with finite flows exponentially distributed.

Regarding (14) and (15), we prove formally only the terms related to the contribute of the variation of the workload on the population $P(w,t)$. 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, 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.

E. Sources with slow start and threshold mechanisms

We report here how an accurate model of sources employing slow start and threshold mechanisms can be developed within our modeling framework.

Let $P(w,t,s)$ be the number of flows with window $\leq w$ and threshold $\leq s$, with $s \geq 1$. Whenever a source experiences a loss signal, it halves its window and sets its threshold equal to the window. In this model, no memory is kept of the previous threshold before the experienced loss. We have to discriminate among three cases, depending on the relation between w and s. We can derive:

$$\begin{aligned}\frac{\partial P}{\partial t}(w,t,s) = &-\frac{1}{R(t)} \int_{\beta=1}^{w} \frac{\partial^2 P}{\partial w \partial \beta}(w,t,\beta) d\beta \\ &- \frac{w}{R(t)} \int_{\beta=w}^{s} \frac{\partial^2 P}{\partial w \partial \beta}(w,t,\beta) d\beta + \\ &+ \int_{\alpha=w}^{2w} \int_{\beta \geq 1} \lambda_{1/2}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta \\ &+ \int_{\alpha=1}^{w} \int_{\beta \geq s} \lambda_{1/2}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta \\ &\text{for } 1 \leq w \leq s \quad (21)\end{aligned}$$

$$\begin{aligned}\frac{\partial P}{\partial t}(w,t,s) = &-\frac{1}{R(t)} \int_{\beta=1}^{s} \frac{\partial^2 P}{\partial w \partial \beta}(w,t,\beta) d\beta \\ &+ \int_{\alpha=1}^{w} \int_{\beta \geq s} \lambda_{1/2}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta + \\ &+ \int_{\alpha=w}^{2s} \int_{\beta=1}^{s} \lambda_{1/2}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta \\ &\text{for } s < w \leq 2s \quad (22)\end{aligned}$$

$$\begin{aligned}\frac{\partial P}{\partial t}(w,t,s) = &-\frac{1}{R(t)} \int_{\beta=1}^{s} \frac{\partial^2 P}{\partial w \partial \beta}(w,t,\beta) d\beta \\ &- \int_{\alpha=2s}^{w} \int_{\beta=1}^{s} \lambda_{1/2}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta + \\ &+ \int_{\alpha=1}^{2s} \int_{\beta \geq s} \lambda_{1/2}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta \\ &\text{for } w > 2s \quad (23)\end{aligned}$$

$$\begin{aligned}\frac{\partial P}{\partial t}(1,t,s) = &-\frac{1}{T_{TO}} P(1,t,s) \\ &+ \int_{\alpha=1}^{2s} \int_{\beta \geq 1} \lambda_{TO}(\alpha,t) \frac{\partial^2 P}{\partial \alpha \partial \beta}(\alpha,t,\beta) d\alpha d\beta \\ &\text{for } s \geq 1 \quad (24)\end{aligned}$$

where T_{TO} is the timeout duration for the TCP sources, that is taken to be constant.

Equation (21) models the case when $w \leq s$. The population $P(w,t,s)$ is decreased by: (i) the sources in congestion avoidance, with threshold less than w and decreasing rate $1/R(t)$ (first term), (ii) the sources in slow start, with threshold between w and s, with decreasing rate $w/R(t)$ (second term). To understand the meaning of the integrals, note that $(\partial^2 P(w,t,s)/\partial w \partial s)$ is the density function of sources with window w and threshold s. $\lambda_{1/2}(w,t)$ is the loss rate given by loss recognition mechanisms (like duplicate acks, etc.) for sources with window equal to w, whose effect is to halve the window and set the new value of the threshold. The population is increased by the sources experiencing the loss signal, in two cases: (i) when the window is between w and $2w$ (third term), (ii) when the window is less than w and the threshold is larger than s (fourth term). Note that the integral with respect to s can be solved analytically for all the four terms and then used in the numerical solver.

Now consider (22), for w between s and $2s$. The population of sources is decreased by all the sources (in congestion avoidance) with threshold $\leq s$ (first term). The population is increased by: (i) sources with window $\leq w$ and any threshold $\geq s$ (second term), (ii) sources with window between w and $2s$ and threshold less than s (third term).

Finally, (23) models the sources with windows $\geq 2s$. The population will be decreased by: (i) sources increasing their window (first term), or (ii) sources with threshold $\leq s$ and window w' experiencing losses (second term), since their new threshold will be set equal to $s' = w'/2 > s$, which is outside the considered population. The population will be decreased only by sources experiencing losses with window $\leq 2s$ and threshold greater than s.

The population with sources at window 1 is described by the boundary equation (24). We assume that the time taken by sources to grow their unit window decays exponentially with average T_{TO}; this explains the first term. The second term is due to sources with window $\leq 2s$ experiencing a timeout; their threshold will be set equal to a value $\leq s$, independently from the initial threshold. Indeed, $\lambda_{TO}(w,t)$ is the rate at which timeouts are experienced; timeouts reset the window to one. Hence, the total loss rate is $\lambda(w,t) = \lambda_{1/2}(w,t) + \lambda_{TO}(w,t)$.

The function λ_{TO}, which models the source reaction to losses, and the occurrence of timeouts, is affected by the version of TCP. One possible choice is given by the approximate model proposed in [16] for which: $\lambda_{TO}(w,t) = \min(1, 3/w) w/R(t) p^F(t)$.