Flow Level Simulation of Large IP Networks

François Baccelli & Dohy Hong
INRIA–ENS
ENS DI, 45 rue d'Ulm, 75230 Paris Cedex 05, France.
Francois.Baccelli@ens.fr, Dohy.Hong@ens.fr

***Abstract*— The aim of this paper is to simulate the interaction of a large number of TCP controlled flows and UDP flows sharing many routers/links, from the knowledge of the network parameters (capacity, buffer size, topology, scheduling) and of the characteristics of each TCP (RTT, route etc.) and UDP flow. This work is based on the description via some fluid evolution equations, of the joint evolution of the window sizes of all flows over a single bottleneck router/link, as function of the synchronization rate. It is shown that the generalization of this fluid dynamics to a network composed of several routers can be described via equations allowing one to simulate the interaction of e.g. millions of TCP flows on networks composed of tens of thousands of links and routers on a standard workstation. The main output of the simulator are the mean value and the fluctuations of the throughput obtained by each flow, the localization of the bottleneck routers/links, the losses on each of them and the time evolution of aggregated input traffic at each router or link. The method is validated against NS simulations. We show that several important statistical properties of TCP traffic which were identified on traces are also present on traffic generated by our simulator: for instance, aggregated traffic generated by this representation exhibits the same short time scale statistical properties as those observed on real traces. Similarly, the experimental laws describing the fairness of the bandwidth sharing operated by TCP over a large network are also observed on the simulations.**

I. Introduction

It is well known that the packet level simulation of TCP over IP with tools like NS2 or Opnet is currently not possible for large populations of flows and/or large numbers of links/routers. For instance, the simulation of the TCP flows of a state of the art access network (say an ADSL or an UMTS access network) is currently unfeasible at packet level. Since these access networks are most often the bottleneck of end to end Internet connections, the elaboration of simulation methods that would allow one to analyze the sensitivity of the main performance metrics w.r.t. the key parameters of such networks (topology, buffer sizes, scheduling and service differentiation strategies) is a challenge of some importance.

Among the main research directions on the simulation of very large IP networks, we would quote parallel simulation, with projects like SSFnet [20], emulation projects like NistNet [17].

The present paper proposes a simplified representation of interacting TCP flows via coupled evolution equations for simulating large IP networks at flow level. The basis of this approach is the AIMD (Additive Increase Multiplicative Decrease) model [4], which describes the joint evolution of the congestion window size of N long lived (FTP type) flows controlled by TCP and sharing a single drop-tail router, in terms of a set of evolution equations. The present paper extends this approach in several complementary and compatible ways:

- General flow models are considered, corresponding to a wide variety of applications: the simulated TCP flows are either long lived (FTP, Peer to Peer) or *on-off* (like in HTTP traffic). They interact with UDP flows;
- Heterogeneous flows can be handled as well: flow can have different Round Trip Times (RTT) or different routes through the network;
- Arbitrary network topologies can be considered, where each flow goes through a route made of several routers/links in series.

The generic model, which is introduced in §II, will be referred to as the multi-AIMD model.

The aim of this model is to allow one to estimate the throughput obtained by each individual flow under the competition rules imposed by TCP, and also the fluctuations of this throughput, from the sole knowledge of the route and the RTT of each flow, and the characteristics of each router and link (buffer size, link capacity, scheduling etc.) in the network.

The simulation is at flow level. It is based on a pathwise description of the dynamics of the interaction between flows, which takes into account discrete event phenomena that are of central importance for tail drop routers/links, such as congestion epochs, losses, synchronization of sources etc. and which allows one to analyze throughput fluctuations.

This new representation of the interaction between TCP flows, which is the main contributions of the present paper, is described in §II and is validated against NS2 simulation in §III-A. We then show in §IV that this method can be used as a simulation tool allowing one to simulate the interaction of large populations of flows on large networks. This is exemplified on the simulation of sizable access networks.

We also show that several important properties of TCP traffic which were identified on traces are also present on traffic generated by our simulator.

We first study the statistical properties of aggregated traffic generated by this representation and we show in §IV-D using wavelet tools that it exhibits the same short time scale statistical properties as observed on real traces [18], [9], [22], [2].

Similarly, the experimental laws describing the bandwidth sharing fairness operated by TCP over a large network are also observed on the simulations.

II. State Variables and Structure of the Simulator

The basic idea of the paper is a decoupling of two different levels, a packet level, which will play a role via the synchronization rate of links and routers, and a flow level rate, that will only retain the synchronization rate from the packet level.

A. State Variables for the Flow Level

The model parameters (exemplified on Figure 1) are the following:

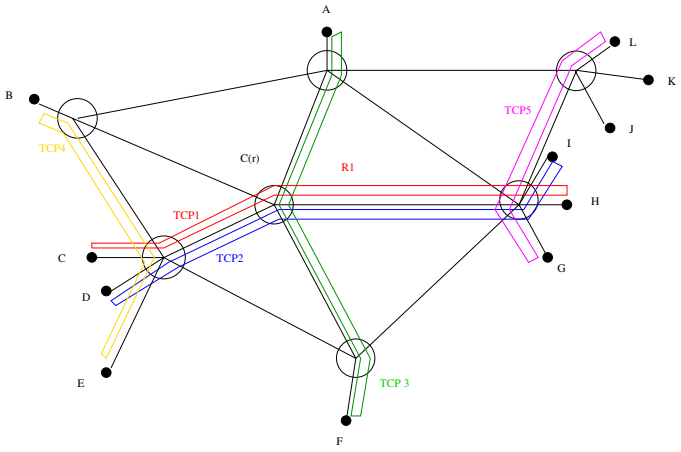

Fig. 1. Several Flows, Several Links & Routers

- Network configuration: $\mathcal{R}$ is the set of routers; C_r is the capacity of router $r \in \mathcal{R}$; B_r is the buffer size of router $r \in \mathcal{R}$; all routers are assumed to be tail drop.
- Traffic configuration: $\mathcal{S}$ is the set of TCP flow classes; N_s is the number of TCP flows of class $s \in \mathcal{S}$; $\mathcal{P}_s$ is the route (forward and backward routes are assumed to be the same) of class s flows; depending on the circumstances, any such route will be considered as a sequence of routers or as a sequence of pairs of routers; $RTT_s = R_s$ is the propagation delay (way and back) for class s flows, which is also the minimal RTT for this class;
- Network and traffic configuration: $\mathcal{S}_r$ is the set of classes with a route using router r;

We now give the notation for the different state variables that we will use. Most of these variables refer to the sequence $\{T_n\}$ of all *congestion epochs* in the network. As in the AIMD model, T_n is the n-th epoch at which a loss (or several simultaneous losses) occur on at least one of the routers/links. As we will see, the simulator constructs all these epochs and the associated state variables step by step from the basic data: network topology, population of flows etc.

- $X^{(s,i)}(t)$ is the throughput of flow i of class s at time t;
- $B_r(t)$ is the buffer content of router/link r at time t;
- $X_n^{(s,i)} = X^{(s,i)}(T_n+)$ is the throughput of flow i of class s just after the n-th congestion time;
- $Y_n^{(s,i)} = X^{(s,i)}(T_n-)$ is the throughput of flow i of class s just before the n-th congestion time;
- $\gamma_n^{(s,i,r)}$ is the multiplicative decrease random variable of flow $i \in s$ on router r at the n-th congestion epoch: $\gamma_n^{(s,i,r)} = 1/2$ if there is a loss for flow i on router r at this epoch and $\gamma_n^{(s,i,r)} = 1$ otherwise; of course $\gamma_n^{(s,i,r)} \equiv 1$ if $r \notin \mathcal{P}_s$.
- $p_n^{(s,r)} = \mathbb{P}(\gamma_n^{(s,i,r)} = 1/2)$ is the *synchronization rate* of router r for the flows of class s at the n-th congestion epoch. We show in §II-D.1 how this synchronization rate can be estimated from the network parameters using simple queueing theoretic arguments.

B. Dynamics in the Simplest Case

In a first step, it will be assumed that routers have no buffer capacity so that it makes sense to assume that the different RTTs are constant over time and equal to R_s for class s. We will see in §II-C how to relax these assumptions that are only made here for the sake of easy presentation.

Assume one knows T_n and $X_n^{(s,i)}$ for all i and s, and that for all $r \in \mathcal{R}$, $\sum_{s\in\mathcal{S}_r}\sum_{i\in s} X_n^{(s,i)} \le C_r$. Due to the Additive Increase (AI) rule, each flow of class s increases its throughput with slope $\frac{1}{R_s^2}$ (this is the slope obtained when assuming that the window size and the RTT are linked at any time by a Little like formula: $W = XR$). So the sum of the throughputs of all flows using router/link r increases with slope $\sum_{u\in\mathcal{S}_r}\frac{N_u}{R_u^2}$. This lasts until the the next congestion epoch T_{n+1}, which is the first epoch after T_n when the sum of the instantaneous throughputs through one of the routers/links exceeds the capacity of this router/link. So we get

$$Y_{n+1}^{(s,i)} = X_n^{(s,i)} + \frac{\min_{r\in\mathcal{R}} \tau_{r,n}}{R_s^2},\ T_{n+1} = T_n + \min_{r\in\mathcal{R}} \tau_{r,n}, \qquad (1)$$

with

$$\tau_{r,n} = \frac{C_r - \sum_{j,u\in\mathcal{S}_r} X_n^{(u,j)}}{\sum_{u\in\mathcal{S}_r}\frac{N_u}{R_u^2}}. \qquad (2)$$

Let $r_n = \text{argmin}_{r\in\mathcal{R}}\tau_{r,n}$. Assume that this set has one element. Then due to the Multiplicative Decrease (MD) rule,

$$X_{n+1}^{(s,i)} = \gamma_{n+1}^{(s,i,r_{n+1})} Y_{n+1}^{(s,i)}. \qquad (3)$$

Should there be several elements in the last set, then one would apply the multiplicative rule for all routers of the set (the order in which the multiplicative decrease is made does not affect the result). We see that these simple rules allow us to compute T_{n+1} and $X_{n+1}^{(s,i)}$ so that we can construct the whole process step by step indeed.

C. Model Refinements

1) General Buffers: the Non-Linear AIMD Model: In the FIFO case, the following evolution equations should be used in-between congestion epochs:

$$\frac{dX^{(s,i)}(t)}{dt} = \frac{1}{(R_s(t))^2},\ R_s(t) = R_s + \sum_{r\in\mathcal{P}_s}\frac{B_r(t)}{C_r}, \qquad (4)$$

$$\frac{dB_r(t)}{dt} = \left(\sum_{s\in\mathcal{P}_r}\sum_{i=1}^{N_s} X^{(s,i)}(t) - C_r\right) 1_{B_r(t)>0} \qquad (5)$$

with R_s the propagation delay (minimal value of RTT) for class s. The evolution equations at congestion epochs are exactly as in the basic Multi-AIMD model. Notice that the slow

start phase can easily be represented by a slight adaptation on the non-linear dynamics.

In this case, the congestion epochs are of course those where one of the buffers overflows. By integrating the last differential equations via a natural discretization, one gets a scheme where one again alternates between non-linear growth periods, and congestion periods where the multiplicative rule is applied.

For further refinements, like for instance more precise differential equations when buffers fill in, see e.g. [7] and [10].

2) HTTP: On-Off Sources: We will limit ourselves to the description of N HTTP sources of the same class sharing a single link/router. The extension to the case with several routes and routers/links is immediate.

Each source brings a potential traffic, which is represented via a sequence of random independent and identically distributed (i.i.d.) *file sizes* and a sequence of random i.i.d. *think times*. Each such source alternates between off periods (the think times) and on periods; the length of each on period depends on the size of the downloaded (or uploaded) file and on the throughput obtained by the source. The number of *on* sources is now a random process $0 \leq N(t) \leq N$.

Source i has a $\{on, off\}$ valued counter and a real valued counter $K^{(i)}(t)$:

- When the source is *off*, $K^{(i)}(t)$, which gives the residual think time, decreases with slope -1; the source jumps from *off* to *on* (birth) when $K^{(i)}(t)$ reaches 0;
- When the source is *off*, $K^{(i)}(t)$ gives the residual number of bits of the file currently under transfer still to be transmitted; in this case, $\frac{dK^{(i)}(t)}{dt} = -X^{(i)}(t)$; the source jumps from *on* to *off* (death) when $K^{(i)}$ reaches 0.

One now defines $\{T_n\}$ to be the sequence of all router/link congestion epochs and all source birth or death epochs interleaved; this sequence is computed step by step during the simulation together with the $\{X_n^{(i)}\}$ state variables.

Between two such epochs, *on* sources follow the same additive increase rule so that their throughputs can be evaluated using the same differential equations as in the long lived case, whereas *off* sources have 0 throughput.

At any such epoch, one updates the individual throughputs and counters as follows:

- If T_n is the birth of source i, then one initializes $K^{(i)}(T_n)$ by sampling a new random file size and $X_n^{(i)} = 0$; all other variables remain unchanged;
- If T_n is the death of source i, then one initializes $K^{(i)}(T_n)$ by sampling a new random think time; all other variables remain unchanged;
- If T_n is a congestion epoch, then for all sources that are *on*, one takes $X_n^{(i)} = \gamma_n^{(i)} Y_n^{(i)}$ where $\gamma_n^{(i)}$ has the same interpretation as in the long lived flow case.

Figure 2 gives an example of trajectory for a HTTP flow as simulated by the non-linear version of these equations. The simulator also implements a simplified version of the slow start.

D. Packet Level and Synchronization

This section is devoted to one possible description of the packet level phenomena, and leads to an estimation of the

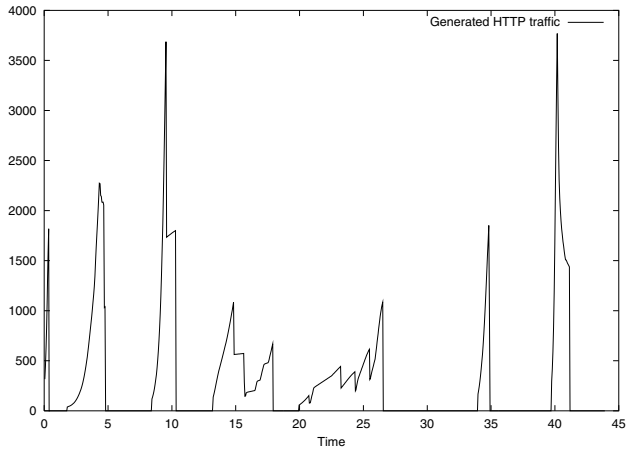

Fig. 2. Example of HTTP trajectory

$\gamma_n^{(s,i,r)}$ variables introduced in §II. As we will see, this allows one to take into account the delay of reaction inherent to TCP. Note that other analytical or simulation based evaluations of the $\gamma_n^{(s,i,r)}$ variables are possible for combination with the flow level equations.

1) Estimation of the Synchronization Rate:

a) Estimation of loss probability at congestion times: In order to estimate the probability that a packet of class s is lost at a congestion epoch of type r, with $r \in \mathcal{P}_s$, we use a simple $M/M/1/B_r$ approximation. In the case of a single link with small buffer, the argument goes as follows. During the congestion period, the total arrival rate is approximately equal to the total service rate, namely C_r. This lasts for a duration of approximately $\eta_r = \min_{s \in \mathcal{S}_r} R_s$ (a quantity that will be referred to as the reaction time for router r), since the flows with the shortest RTT then react to losses on router r, thus ending the congestion period. For the (multiclass) $M/M/1/B_r$ queue with total arrival rate λ and service rate μ such that $\lambda = \mu = C_r$, the steady state packet loss probability is $\underline{L_r} = 1/(B_r + 1)$ regardless of the class. If $B_r/C_r << \eta_r$, it makes sense to approximate the empirical frequency with which packets of any type s, with $r \in \mathcal{P}_s$, are lost on $[0, \eta_r]$ by this stationary probability. In a refined model, we use the same argument but with $\lambda = C_r'$ and $\mu = C_r$, with C_r' the arrival rate into the buffer at the time of congestion, a quantity which can be estimated by the simulator from the current values of the throughputs when the buffer is full. This leads to the following formulas :

$$L_r = \frac{\rho_r^{B_r}(\rho_r - 1)}{\rho_r^{B_r+1} - 1}, \tag{6}$$

where $\rho_r = C_r'/C_r$. We have $L_r \sim \overline{L}_r = \frac{\rho_r^{B_r}}{B_r+1}$ when $\rho_r \to 1$ and $\underline{L}_r = L_r = \overline{L}_r$ when $\rho_r = 1$.

b) Estimation of the synchronization rate: We now propose an estimation of the synchronization rate also based on the $M/M/1/B_r$ queue analysis and when assuming that $B_r/C_r << \eta_r$ and that the population of each class is large. In order to compute the synchronization rate of type r for flows of class s, with $r \in \mathcal{P}_s$, we have to evaluate how many *flows*

of this class experience loss during this congestion period, while taking into account the fact that if a given flow has already experienced a loss, then any further packet loss of this flow that takes place in the very same congestion period should not be counted as a new flow loss. At the beginning of the n-th congestion period, the loss rate of flows of type s coincides with the packet loss rate for this class and is equal to $\sum_{i\in s} Y_n^{(s,i)} L_r$. Let $y_n^{(s)}$ be the empirical mean of the throughputs of flows of class s: $y_n^{(s)} = \frac{1}{N_s}\sum_{i\in s} Y_n^{(s,i)}$. So, the loss rate of flows of type s is $N_s y_n^{(s)} L_r$. After the first loss of this class took place, the flow loss rate becomes approximately $(N_s - 1)y_n^{(s)} L_r$ provided N_s is large (see [5] for a justification); similarly, after the second flow loss, the flow loss rate is approximately $(N_s - 2)y_n^{(s)} L_r$, etc. In order to determine the mean number of flows of class s that experience at least one loss by time η_r, we have to study the transient mean value of the continuous time pure birth process on the integers with birth rates $\lambda_{i,i+1} = (N_s - i)y_n^{(s)} L_r$. Doing so, we obtain that the synchronization rate for the flows of class s at time T_n can be estimated as $p_n^{(s,\overline{r}_n)}$, with the function $p_n^{(s,r)}$ given by the formula

$$p_n^{(s,r)} = \frac{1 - e^{-y_n^{(s)}\eta_r L_r}}{1 - e^{-C_r \eta_r L_r}}. \qquad (7)$$

The proof of this formula is forwarded to the appendix.

Notice that for this formula to be valid, it is necessary that for each router/link r, the mean time between congestions of this resource be larger than η_r.

In order to test the accuracy of Formula (7), we compared the performance result with NS simulations. For a single router bottleneck case with tens of parallel sessions, NS simulations give performance results that have a variation of 10 to 20% when changing parameters others than the capacities and propagation delays. This variation seems to decrease when the number of flows increases, and also when timeouts are negligible. Our performance prediction using the formula (7) is within the range of results given by NS. For more on the matter, see §III-A.

2) Rate-Dependent Losses: For a rate-dependent synchronization stochastic model, the probability that the random variable $\gamma_n^{(s,i,r)}$ is 1/2 is an increasing function of the throughput of source i just before the n-th congestion time, that is $Y_n^{(r,s,j)}$, $j \in s$, $r \in \mathcal{P}_s$.

III. Comparison with NS

A. Validation

In this section, a partial validation of this flow level approach is made against NS2. Table 1 summarizes the results for the FIFO single link, long lived flow case. This table reports on the non-linear AIMD simulator, based on the estimation of synchronization of (7) and (6). More precisely, the synchronization rate is dynamically evaluated and is exponentially proportional to the sendrate and the reaction delay. In this table, mrtt denotes the mean RTT as obtained from Equation (4). TD/TO gives the ratio of the frequency of losses (Triple Duplicates or TD) and of that of timeouts (TO). Simtime denotes the simulated time. The NS2 simulation results very much depend on certain external parameter, in particular the end-users link speed: indeed this speed modifies the inter-packet distance inside RTT, which has a crucial impact on the synchronization rate. In the tables the NS2* entries that follow a NS2 entry exemplify the effect of variations of the speed of these links. They give values obtained when keeping all parameters as in the associated NS2 entry, but for the end-users link speed for which we take values ranging from 1000 Mb/s to 100/10/5 Mb/s.

In all cases, the rates (input or output) that are given by the AIMD simulation are within the range of the NS2 results. The AIMD simulation results are very accurate except when the timeout probability becomes large (TO rate/TD rate $> 10\%$, cf. Table 1 with 50 sessions).

As it was already indicated, Formula (7) which is used here is only valid when the mean intercongestion time is larger than the RTTs. We observed that this last condition was satisfied in all cases with $E[w] > 2$. Therefore, when timeouts are not dominant, our model is consistent and robust.

This model is fine for more than mean values. Trajectories are quite close too, at least in the case with few timeouts. Figure 3 plots NS2 and AIMD trajectories in the case of the 1st line of the table over 500 seconds.

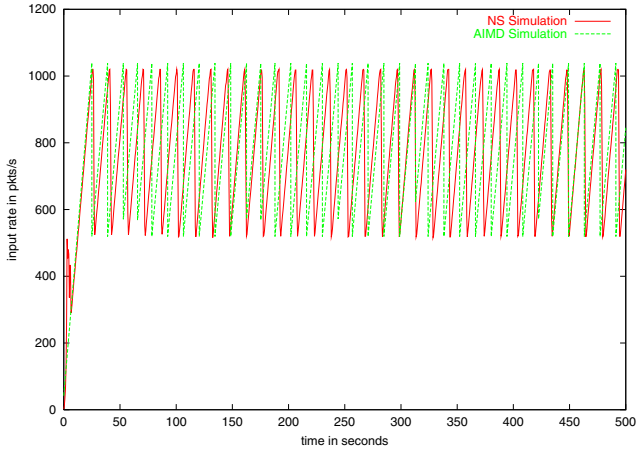

Fig. 3. First line of the table

We now move to the validation on the three–class, two–router network of Figure 4. As above, NS2* studies the variations w.r.t. the external link speeds (800/80/8 Mb/s).

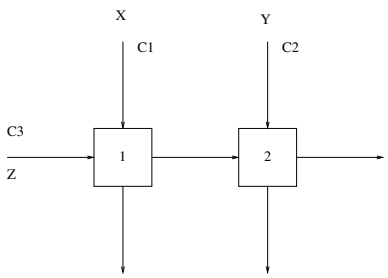

Fig. 4. 2 Router, 3 Class Network Topology

# sess.	rttmin	BW	Buff.	tool	input/output	mrtt	TD/TO	Simtime	runtime	$\frac{\text{NS}}{\text{AIMD}}$
	ms	pkts/s	pkts		pkts/s	ms	%	s	min:s	runtime
10	500	1000	20	NS2	771/770	500	0.11/4e-4	1000	1:20	
.	.	.	.	NS2*	784/783	500	0.11/9e-4	1000	1:21	
.	.	.	.	NS2*	837/836	502	0.14/7e-4	1000	1:23	
.	.	.	.	AIMD	784/783	501	0.10/-	1000	0:09	9
10	200	1000	20	NS2	814/810	202	0.43/1e-4	1000	1:22	
.	.	.	.	NS2*	863/859	203	0.48/5e-4	1000	1:27	
.	.	.	.	NS2*	898/893	204	0.54/6e-4	1000	1:24	
.	.	.	.	AIMD	844/839	202	0.53/-	1000	0:09	9
30	200	3000	100	NS2	2576/2564	204	0.42/5e-4	1000	3:31	
.	.	.	.	NS2*	2849/2833	206	0.51/6e-4	1000	3:43	
.	.	.	.	AIMD	2575/2562	204	0.51/-	1000	0:09	21
50	200	1000	40	NS2	935/899	210	3.8/0.3	1000	4:04	
.	.	.	.	NS2*	992/946	210	4.4/0.5	1000	4:05	
.	.	.	.	NS2*	1029/983	212	4.4/0.5	1000	4:09	
.	.	.	.	AIMD	999/918	223	8.1/-	1000	0:10	24
100	200	10000	200	NS2	8227/8189	202	0.43/4e-4	1000	11:01	
.	.	.	.	NS2*	8567/8524	202	0.47/9e-4	1000	11:16	
.	.	.	.	AIMD	8540/8495	202	0.53/-	1000	0:10	66
500	200	100000	100	NS2	58695/58405	200	0.17/0.03	100	8:55	
.	.	.	.	AIMD	84942/84831	200	0.13/-	100	0:01	540
10000	500	100000	1000	AIMD	90014/87832	501	2.4/-	1000	0:53	

Table 1:AIMD against NS2 – Single Link Case

Case 1, studied in Table 2, is with C1=C2=1000 pkts/s and B1=B2=20 pkts.

There is still high sensitivity of NS2 simulation results w.r.t. parameters that impact on the inter-packet distance within RTTs. Case2, studied in Table 3, is with C1=4000 pkts/s, C2=2000 pkts/s and B1=B2=20 pkts.

Figure 5 shows the evolution of the aggregated throughput (normalized by the number of flows) as obtained by NS2 simulation and by our AIMD simulation. Figure 6 is a zoom on class 3 traffic. From this, we conclude that both in the single link and the multiple link cases, the flow level simulation gives results that are within the range of variations of those of NS2, provided timeouts are rare enough. This is true for mean values and also for more qualitative features such as the shape of trajectories.

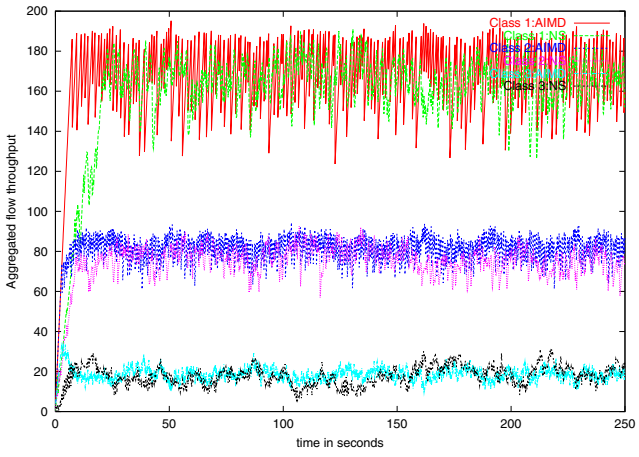

Fig. 5. Aggregated Throughput

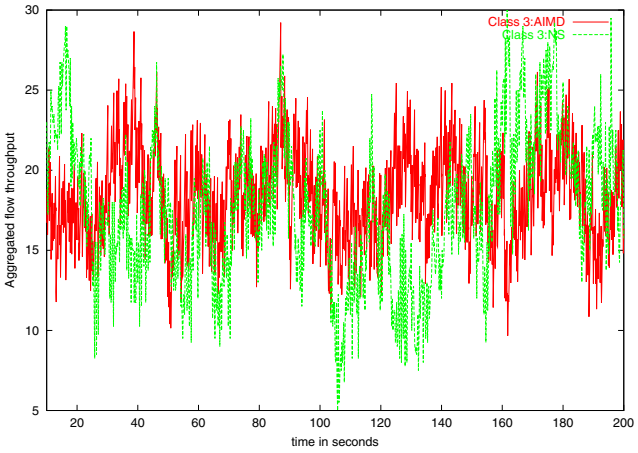

Fig. 6. Zoom on Aggregated Throughput

B. *Packet Burstiness and High Speed Links*

The fact that NS2 simulation results depend very much on the local link speeds of the sources (variations up to 300% in Table 2) is mainly due to the fact that the packet distribution inside one RTT may have a huge impact on the synchronization and on the performance. When each these speeds are not too large, our fluid model are accurate.

In the case of high speed links (Table 1, C=100000 pkts/s

tool	class	# sess.	rttmin	input/output	mrtt	TD/TO	Simtime	runtime	$\frac{\text{NS}}{\text{AIMD}}$
			ms	pkts/s	ms	%	s	min:s	runtime
NS2	1	10	200	514/507	210	1.4/0.01	500	1:03	
	2	10	200	535/528	211	1.4/0.008	.	.	
	3	10	200	425/417	219	2.1/0.07	.	.	
NS2*	1	.	.	805/800	211	0.61/7e-4	500	1:03	
	2	.	.	817/813	211	0.54/1e-3	.	.	
	3	.	.	154/141	245	8.2/1.6	.	.	
AIMD	1	.	.	558/551	202	1.2/-	500	0:14	4.5
	2	.	.	555/548	203	1.2/-	.	.	
	3	.	.	341/330	205	3.3/-	.	.	

Table 2:AIMD against NS2 – Two Link, Three Class Case

tool	class	# sess.	rttmin	input/output	mrtt	TD/TO	Simtime	runtime	$\frac{\text{NS}}{\text{AIMD}}$
			ms	pkts/s	ms	%	s	min:s	runtime
NS2	1	20	200	3274/3267	200	0.25/3e-4	1000	4:32	.
	2	20	200	1535/1522	204	0.84/4e-3	.	.	.
	3	20	300	376/355	305	5.5/0.59	.	.	.
NS2*	1	.	.	3295/3288	200	0.22/1e-3	1000	4:45	.
	2	.	.	1568/1556	204	0.79/6e-3	.	.	.
	3	.	.	334/312	307	6.6/0.86	.	.	.
AIMD	1	.	.	3219/3213	201	0.17/-	1000	0:15	.
	2	.	.	1546/1534	204	0.73/-	.	.	.
	3	.	.	355/334	304	6.0/-	.	.	18

Table 3:AIMD against NS2 – Two Link, Three Class Case

or equivalently 800 Mb/s, 500 sessions), NS2 shows a very bad performance. This is due to the fact that with high speed local links, packets are very likely to be concentrated at the beginning of RTTs. Such a packet concentration creates losses even if the input rate averaged over one RTT is much smaller than the capacity of the shared resource. This is illustrated by Figure 7, where we compare the trajectories obtained with NS2 simulation (where packet concentration takes place) with those of a fluid model where packets are well spread out over the RTT by construction. When averaging the input rate over RTT=0.2 s, we see that losses occurs very frequently even if the input rate is far from the 800 Mb/s capacity of the shared resource. The reason is clearly shown by the plots of the input rate averaged over 0.1s. Such a burstiness inside a RTT is clearly very negative for performance, in particular for high speed connections.

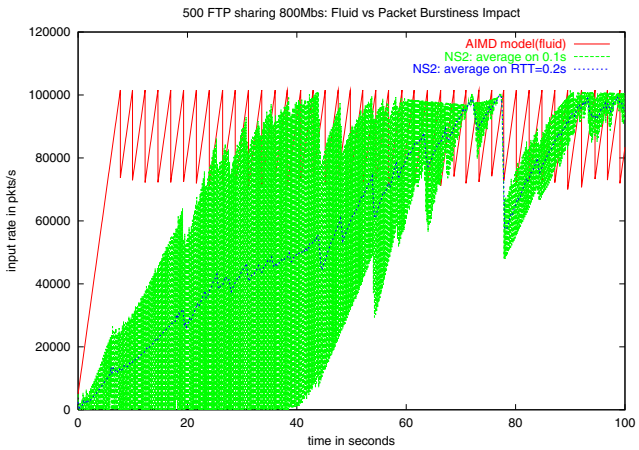

Fig. 7. The negative effect of packet concentration

For a single router/link of capacity C, shared by N FTP users with the same RTT, the fluid model (which describes the situation where packets are ideally 'paced' inside each RTT) predicts that the mean throughput obtained by each user is at least 75% of the ideal fair share C/N (cf. [4]).

The fact that with NS simulations or in real experimentations one may observe a degradation of performance higher than that (for FTP sessions) is mainly due to this packet burstiness inside the RTTs and this effect has an influence on the performance roughly proportional to the concentration rate of packets in RTT.

IV. Case Studies

The simulator described in the previous sections is now used for a few case studies. For these case studies, the non-linear AIMD model (§II-C) is used; it allows one to take the most important effects into account, including, buffer contents tracking, the effect of buffer size on RTT, the slow start, the delay in reaction etc. The aim of this section is twofold.

- We first show that this simulator allows one to study fine properties of large networks, including the sensitivity

of throughput w.r.t various network parameters. This is possible because the simulation cost is approximately linear in the number of the congestion epochs and also in the number of TCP flows.

- We then study the statistical properties of the aggregated traffic generated by this model and compare them to what is reported in the literature.

A. Network Topology

The network topology that is studied is featured on Figure 8 is a simplified model for an access network. In addition to the hierarchical traffic (which will be referred to as the main traffic below), we often add some cross-traffic flows to routers of certain levels. By definition, a cross traffic flow uses this specific router only. By aggregated traffic of level k, we understand the sum of the throughputs of all sources that use a typical router of level k, divided by the total number of such sources.

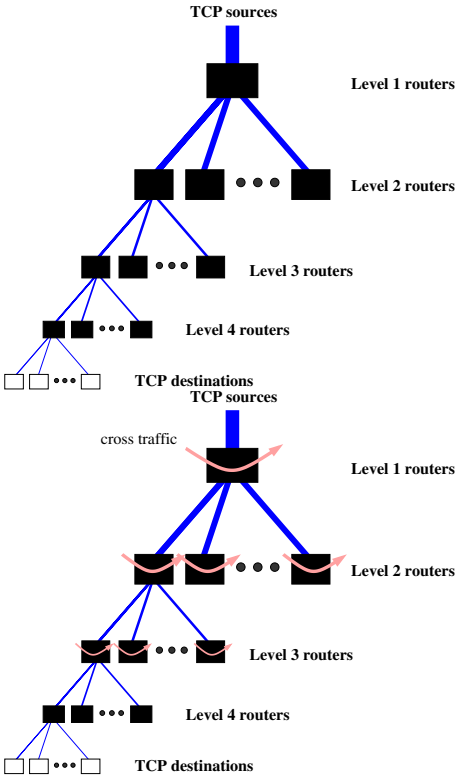

Fig. 8. Tree Topology. Top: without cross traffic. Bottom: with cross traffic.

B. Bottleneck Analysis

The simulator shows that for most configurations, and in particular for configurations as those depicted in Figure 8, there are more than one bottleneck router (or here bottleneck level) for a given flow. For this, the relevant variables are the proportions of congestion epochs (bottlenecks), and the proportion of losses (MD's), that are of a given type (or level) over time in the stationary regime. The two should be distinguished because of the synchronization rates: a bottleneck or congestion epoch at level 1 might create a huge number or losses or MD's even with a moderate synchronization rate, which is not the case at higher levels.

In some particular cases, such as the case where all RTTs are exactly the same and where in addition, there is no cross traffic, a single router approximation could possibly be used. But even in this case, when varying capacities, the transition of the bottleneck from one level to another is not instantaneous. The stationary proportions in question are plotted for this case on the left part of Figure 9, which shows the variations of these proportions when increasing the service capacity of the level 2 routers (which is the bottleneck level on the left part of the plot). In this case, the network is a three level tree. Each router of level 3 is an access router with 10 long lived TCP flows and has a capacity C_3. Level 2 routers are concentrating the flows from 20 routers of level 3 and have a capacity of $C_2 = 10$ Mb/s. The level 1 router concentrates the flows of 30 routers of level 2 and has a capacity of $C_1 = 300$ Mb/s. When $C_3 = 450$ Kb/s, the leaves of the tree are the bottleneck. The transition of the MD proportion curves is rather fast: with a variation of 1.4% of C_3, the MD proportion varies from 100%-0% to 50%-50%. The transition of the bottleneck curves requires an increase of 20 % of the initial bottleneck capacity C_3 (from the value 500 Kb/s).

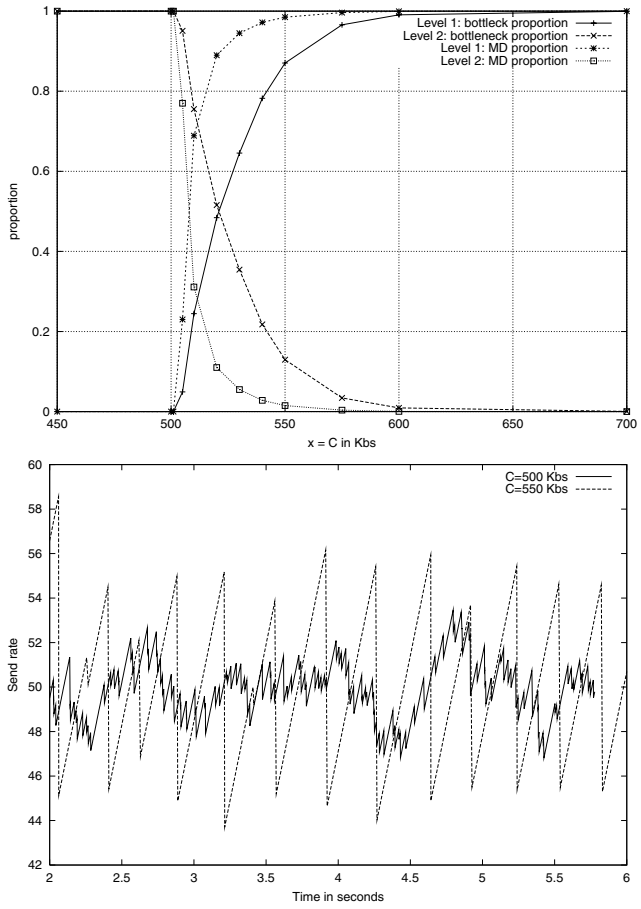

Fig. 9. Bottleneck Transition. Top: loss and bottleneck proportions; Bottom: Level 2 aggregation of TCP flows.

This implies that aggregated traffic seen from Level 2 has

statistical properties that are very sensitive w.r.t. capacity characteristic (cf. the 2nd curve of Figure 9).

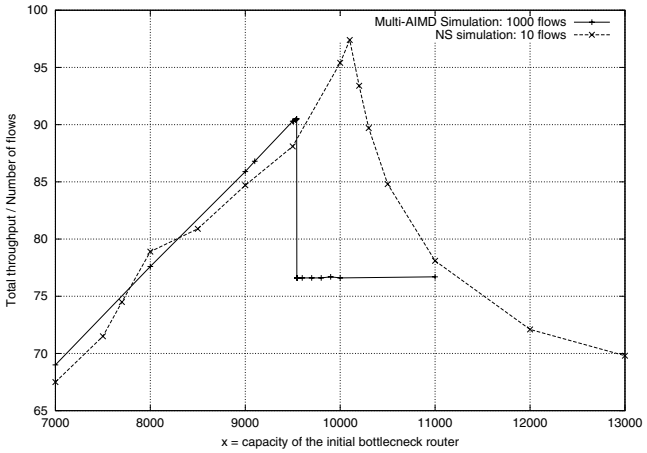

Fig. 10. Non-monotonicity due to synchronization effect.

Figure 10 provides a very simple example of situation where the synchronization rate could introduce some unexpected throughput behavior. In this case, we have a single bottleneck router r shared by N flows and we choose its buffer size and the RTTs of the flows such that the synchronization rate at this router is small. We then connect router r to a second router r' (on the common route of all flows) with a very small buffer size and with a speed 30% bigger than that of R_2. Keeping everything unchanged, if one increases the speed of router r, one should observe a transition of the bottleneck and when r' becomes bottleneck; surprisingly enough, what we observe when doing so is actually a throughput drop. This is actually observed both on a NS simulation and on the AIMD simulation. In the NS simulation we checked that the drop is not due to timeouts but actually to the high synchronization.

C. Sensitivity w.r.t. RTT's

Consider a two level hierarchical network within the class described above, with bottlenecks at two different levels: (local) bottleneck routers at level 2, each with a group of 100 flows, and a global bottleneck at level 1, which concentrates 50 level 2 routers, that is 5000 flows. This time however, RTTs are heterogeneous (sampled uniformly from 1 ms to 2 s). The analysis of the proportions of losses per class obtained from the simulator shows that slow flows (with large RTTs) are less affected by the local bottleneck, whereas the fast ones are mainly affected by it. Figure 11 gives a log-log plot of the mean throughput as a function of RTT. This figure shows that, as in the single router case, the mean throughput $E[X]$ is linked to the RTT by an empirical rule of the form $E[X] \sim K(\text{RTT})^a$, which is consistent with results in e.g. [13], [16]. Our simulations also show that within a group of fast flows, $a \sim 2$, whereas within a group of slow flows, $a \sim 1$. In this multi router case, the transition between these two groups seems to be more progressive than in the single router case.

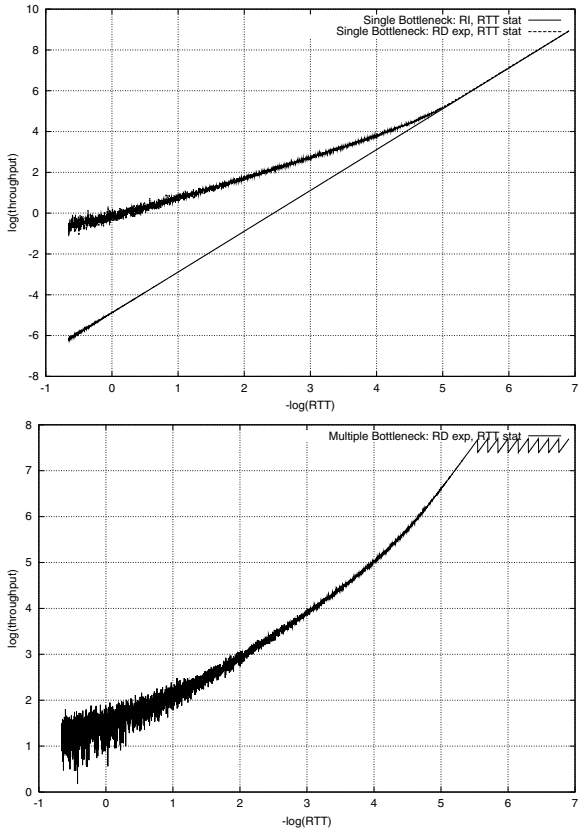

Fig. 11. Throughput vs RTTs in log-log scale. Top: single router, comparison of RI (rate-independent) and RD (rate-dependent) loss models; Bottom: 2 level multi router case with bottlenecks at each level.

D. Aggregated Traffic Analysis

In this section, the network is a 4 level tree. The number of routers of level $n+1$ is 10 times the number of routers of level n. In Case 1, the routers capacities are equal to 500 Mb/s, 50 Mb/s, 5 Mb/s, 0.5 Mb/s, and the buffer capacities are equal to 10000, 1000, 100, 10 packets respectively. For Case 2, the capacity of the last level (leaf) routers is increased to 1 Mb/s, the buffer capacities are modified to 6000, 2000, 100, 10 respectively and cross traffic is added. Each cross traffic is made of long lived TCP flows and is local to each router. This additional traffic is present on routers of all levels (but for level 1), and consists of an additional number of users that amounts to 10% of the main traffic going through this router. We generated 4 classes of propagation delays (RTTmin): 0.1, 0.2, 0.3 and 0.4 seconds, with equal probability, the mean RTT seen by one flow being then approximately equal to RTTmin plus 0.35-0.45 s.

1) Fluctuation and Bottleneck Analysis: The simulation results for Case 1 are given in Figure 12. The top curve concerns the case when losses are at the leaves of the tree (all of them in this case). When aggregating a larger number of flows, the fluctuations decrease as predicted by the law of large number. The bottom curve gives both the transient and the stationary parts of the aggregated throughputs. The convergence to stationary regime is most often exponentially fast, which is consistent with the observations in [6].

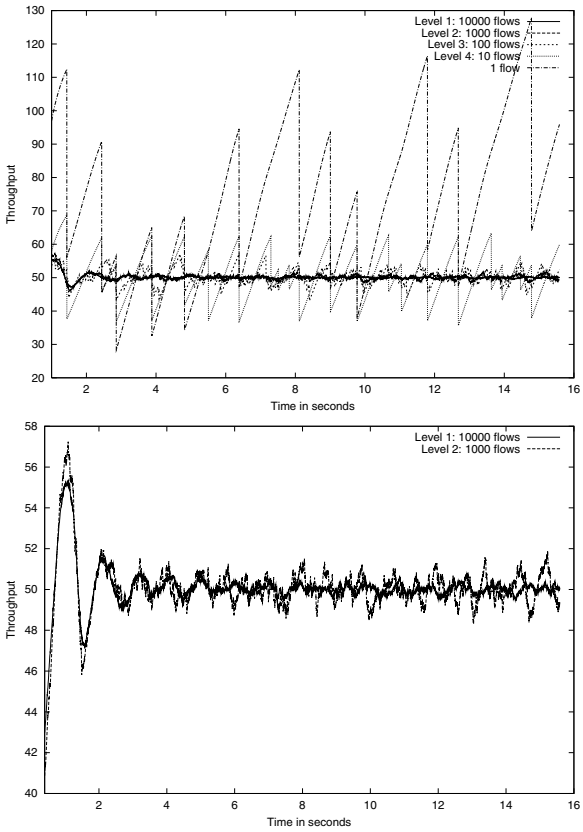

Fig. 12. Throughput evolution over time. Top: Case 1, the 5 levels of aggregation (from 1 to 10000 flows). Bottom: Case 1, the geometric convergence of aggregated traffic to its stationary regime.

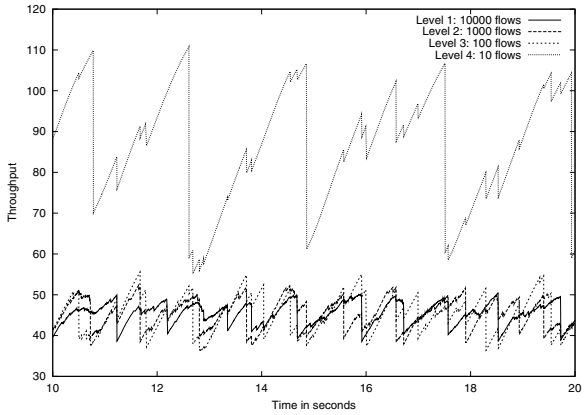

Fig. 13. Case 2: Aggregation at different levels of the tree.

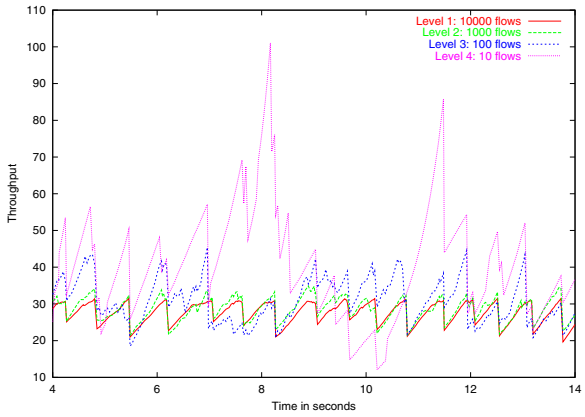

Fig. 14. HTTP case: Aggregation at different levels of the tree.

Figure 13 features Case 2. In this case, losses are present at all levels of the tree and fluctuations are not erased by aggregation, even if level 1 is very rarely bottleneck. In this case, 80.4% of bottlenecks are at level 4, 17.4% at level 3, 2% at level 2 and 0.2% at level 1. The respective mean synchronization rates are 0.50, 0.39, 0.28 and 0.22. The mean throughput averaged over the 4 classes is 35 Kb/s. The value of C_r/N_r (router capacity divided by the number of flows sharing this router) is 45, 45, 45 and 90 Kb/s for levels 1,2,3 and 4 respectively. Therefore the global under-utilization is of 22% (more precisely $1 - 35/45$). This is much more than what the single router AIMD model would predict from the value of the synchronization of the 4 levels: using the formula derived in [4], we would get the following values for under-utilization: $p/4 = 12.5, 10, 7, 5.5\%$ for the various levels. This shows that even in tree like networks, the presence of multiple bottlenecks creates phenomena that cannot be approached by the analysis of the single dominant bottleneck.

Figure 14 studies a configuration similar to that of Case 1 but with traffic of the HTTP type as described in §II-C.2. Here, the main bottleneck is at level 1, and one also observes the absence of statistical multiplexing when agregating flows.

Figure 15 studies that case when the topology of the network is a 4 level tree with an additional cross traffic. In this case, the total number of TCP flows is 1210000 and there are 10211 routers.The routers characteristics are then as follows:

- Level 4: $C = 6$ Mb/s, $B = 100$ pkts shared by 100 flows of the main traffic;
- Level 3: $C = 250$ Mb/s, $B = 5000$ pkts shared by $100 \times 50 = 5000$ flows of the main traffic and 500 cross flows;
- Level 2: $C = 5$ Gb/s, $B = 100000$ pkts shared by $100 \times 50 \times 20 = 100000$ flows of the main traffic and 10000 cross flows;
- Level 1: $C = 50$ Gb/s, $B = 1000000$ pkts shared by $100 \times 50 \times 20 \times 10 = 1000000$ flows of the main traffic and 100000 cross flows.

In this case, losses necessarily occur at each level due to the presence of cross traffic. Taking capacities (buffer and speed) of each level proportional to the number of flows at this level leads to a synchronization rate that does not vary too much from level to level: 0.48, 0.46, 0.36, 0.45. As already noticed in Case 1, the lower levels of the tree are less often bottleneck: we get here 0.02, 0.19, 4.4 and 95 % from level 1 to 4.

2) Distribution Function of Aggregated Throughput: Fig. 16 gives the empirical distribution function for aggregated traffic at each level. The distribution functions exhibit quite different shapes (see e.g. the level 1 with two peaks compared to the more Gaussian like distribution of level 4).

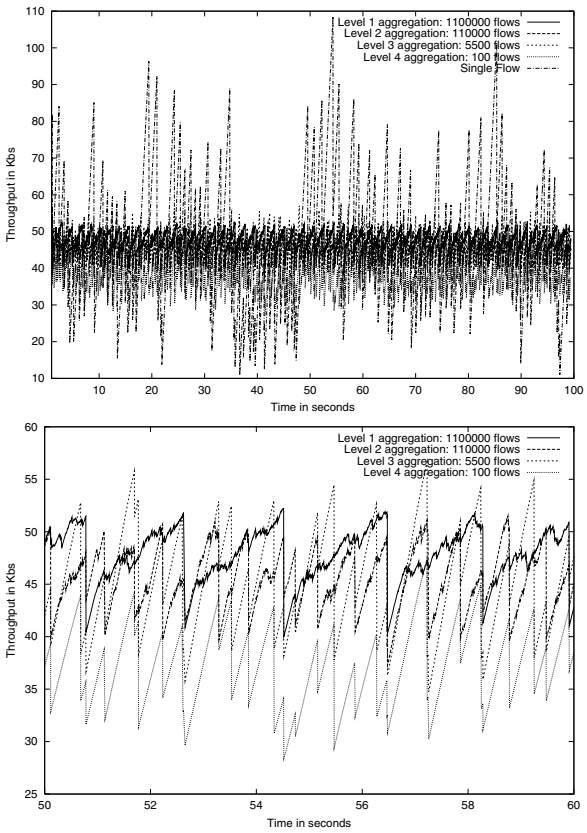

Fig. 15. Throughput evolution. Top: the 5 aggregation levels (from 1 to 1000000 flows); Bottom: Zoom on 4 levels.

3) Scaling Properties: We tested the statistical properties of traffic aggregated at different levels using the Matlab tool developed by P. Abry and D. Veitch [1]. Figure 17 gives the second order logscale diagram (LD) of the energy function and Figure 18 the multiscale analysis (MS) diagram of aggregated traffic (for more on these diagrams, see e.g. [2]). The different levels exhibit statistical properties for LD and MS plots which are similar to those observed in [4] for the single router case, and which are compatible with a multi-fractal scaling.

Observe that whereas the empirical distribution functions we obtained exhibit quite different shapes, the LD and MS analysis are quite insensitive w.r.t. the level of aggregation. The scaling exponent α is between 1.83 and 1.96 in all cases.

V. Appendix: Proof of Formula (7)

Let $N_s(t)$ be the pure birth process described just before Formula (7). It admits the stochastic intensity $\lambda(t) = y_n^{(s)} L_r (N_s - N_s(t))$. So, from the stochastic integration formula (see e.g. [3]),

$$\begin{aligned} E[N_s(t)] &= E\left[\int_0^t N_s(du)\right] \\ &= E\left[\int_0^t y_n^{(s)} L_r (N_s - N_s(u)) du\right] \\ &= y_n^{(s)} L_r N_s t - y_n^{(s)} L_r \int_0^t E[N_s(u)] du. \end{aligned}$$

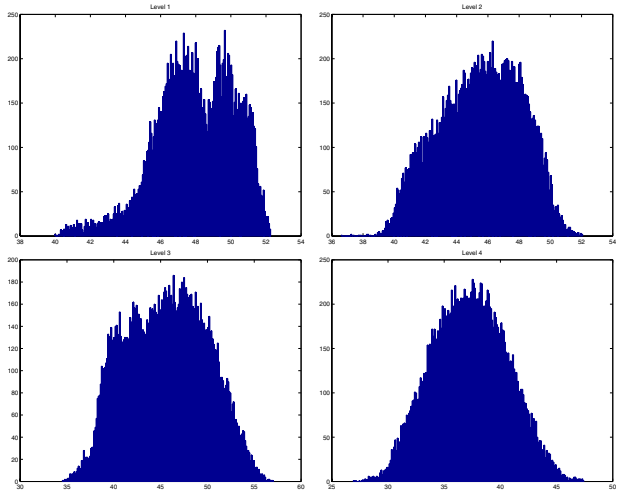

Fig. 16. Distribution Function for Aggregated Traffic at Levels 1-4

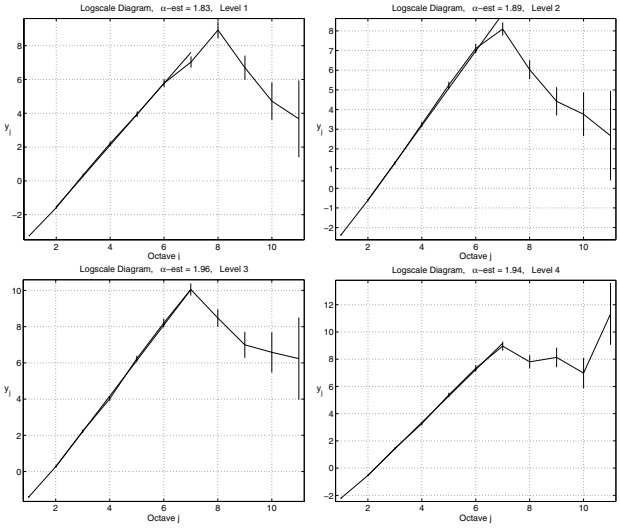

Fig. 17. LD Plots for Levels 1-4.

So, $g(t) = E[N_s(t)]$ satisfies the differential equation $g'(t) = y_n^{(s)} L_r N - y_n^{(s)} L_r g(t)$, with initial condition $g(0) = 0$. The solution is $g(t) = N_s(1 - \exp(-y_n^{(s)} L_r t))$. By the same argument, the expected number of flows that experience at least a loss given that this number is positive is $h(t) = N_s(1-\exp(-y_n^s L_r t))/(1-\exp(-C_r L_r t))$. So the proportion of flows of class s that experience at least one loss by time η_r given that at least one flow looses is as given in the formula.

VI. Conclusion

We have introduced a new flow level simulation method allowing one to study the bandwidth sharing operated by TCP on networks composed of several routers. This approach was shown to provide an efficient framework to simulate large networks. The results obtained by this flow level simulator take into account key packet level phenomena such as the reaction delay, the scheduling and the buffer overflows, via the estimate used for the synchronization rate. The performances

of interacting flows obtained by this approach are in the range of results obtained by NS2 simulations. The experiments based on this simulation technique lead to functional dependencies between throughput and RTT that are compatible with recent observations, and to statistical properties for short time scales that were observed on real traces. The main interest of this simulator stems from its ability to handle very large networks and populations.

REFERENCES

[1] Abry, P. and Veitch, D. http://www.emulab.ee.mu.oz.au/∼darryl

[2] Abry, P., Flandrin, P., Taqqu, M.S. and Veitch, D. (2000) Wavelet for the analysis, estimation and synthesis on scaling data. *Self Similar Traffic Analysis and Performance Evaluation*, Park, K. and Willinger, W. Eds, Wiley.

[3] Baccelli, F. and Brémaud, P. (1994) Elements of Queueing Theory, Springer Verlag.

[4] Baccelli, F. and Hong, D. (2002) AIMD, Fairness and Fractal Scaling of TCP Traffic. *Proc. of INFOCOM*, New York, June.

[5] Baccelli, F. and Hong, D. (2003) Interaction of TCP Flows as Billiards, *Proc. of INFOCOM*, San Francisco, April.

[6] Bohacek, S. Hespanha, J. P., Lee, J. and Obraczka K. (2001) A Hybrid Systems Framework for TCP Congestion Control, *Technical Report*, USC, Los Angeles, CA, July.

[7] Bonald, T. (1998) Comparison of TCP Reno and TCP Vegas via Fluid Approximation. *Technical Report*, RR-3563, INRIA Sophia-Antipolis.

[8] Bonald, T. and Massoulié, L. (2001) Impact of fairness on Internet performance. *ACM SIGMETRICS*, pp. 82-91.

[9] Feldmann, A., Gilbert, A.C., Huang, P. and Willinger, W. (1999) Dynamics of IP traffic: A study of the role of variability and the impact of control. *Proc. of ACM-SIGCOMM'99*, August-September, Cambridge, MA, pp. 301-313.

[10] Hong, D. (2003) A Note on the TCP Fluid Model. *Technical Report*, INRIA Rocquencourt, to appear.

[11] Hong, D. and Lebedev, D. (2001) Many TCP User Asymptotic Analysis of the AIMD Model. *Technical Report*, RR-4229, INRIA Rocquencourt.

[12] Hurley,P., Le Boudec, J.Y., Thiran, P. (1999) A Note on the Fairness of Additive Increase and Multiplicative Decrease. *Proc. of ITC-16*, Edinburgh, June.

[13] Lakshman, T.V., Madhow, U. (1997) The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. on Networking*, 5-3, pp. 336-350.

[14] Massoulié, L. and Roberts, J. (1999) Bandwidth sharing: objectives and algorithms. *Proc. of INFOCOM*, New York.

[15] Mathis, M., Semske, J., Mahdavi, J. and Ott T. (1997) The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *Computer Communication Review*, 27(3), July.

[16] Mo, J. and Walrand, J. (2000) Fair End-to-End Window-based Congestion Control. *IEEE/ACM Trans. on Networking* 8-5, pp. 556-567.

[17] Nistnet, http://snad.ncsl.nist.gov/itg/nistnet/.

[18] Riedy R. and Levy-Vehel, J. (1996) Multifractal Properties of TCP Traffic. *Technical Report*, RR-3129, INRIA Rocquencourt.

[19] Roberts, J. and L. Massoulié. (1998) Bandwidth sharing and admission control for elastic traffic. *ITC Specialist Seminar*, Yokohama, October.

[20] Ssfnet, http://www.ssfnet.org.

[21] Vojnovic, M., Le Boudec, J.Y., Boutremans, C. (2000) Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times. *Proc. of IEEE INFOCOM*, Tel Aviv.

[22] Willinger, W., Paxson, V. and Taqqu, M.S. (1998) Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic. *A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy Tailed Distributions*, R. Adler, R. Feldman and M.S. Taqqu (Eds.), Birkhauser Verlag, Boston, MA, pp. 27-53.

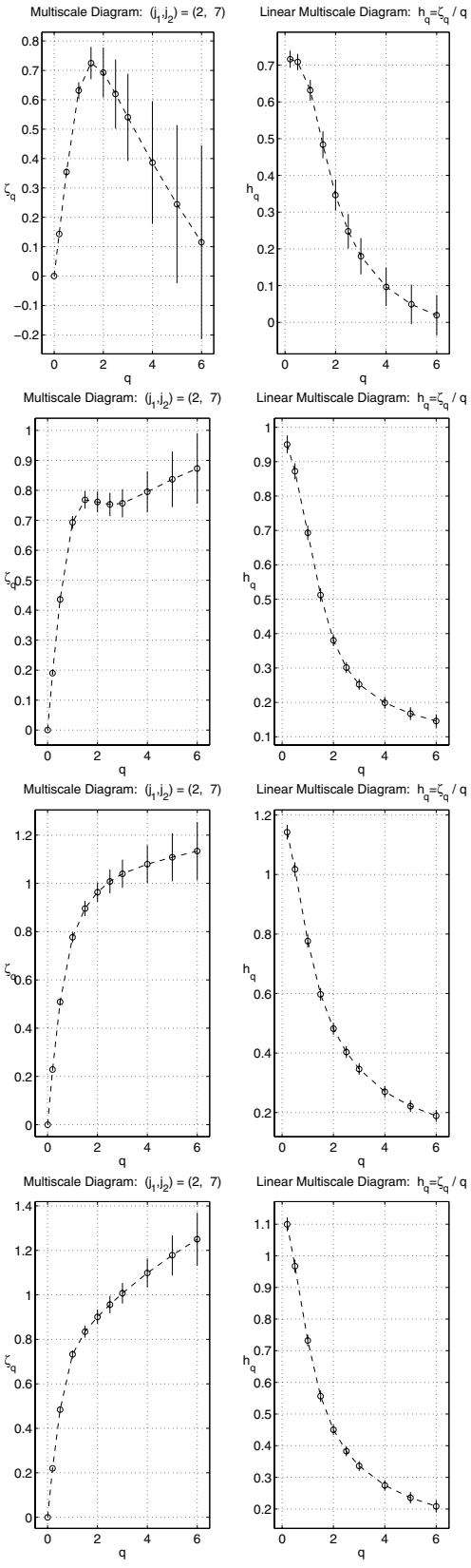

Fig. 18. Multiscale Analysis for Levels 1-4.