Fixed Point Methods for the Simulation of the Sharing of a Local Loop by a Large Number of Interacting TCP Connections

François Baccelli, Dohy Hong, Zhen Liu

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Fixed Point Methods for the Simulation of the Sharing of a Local Loop by a Large Number of Interacting TCP Connections

François Baccelli — Dohy Hong — Zhen Liu

N° 4154

Avril 2001

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4154--FR+ENG

Fixed Point Methods for the Simulation of the Sharing of a Local Loop by a Large Number of Interacting TCP Connections

François Baccelli [*] , Dohy Hong [†] , Zhen Liu [‡]

Thème 1 —Réseaux et systèmes
Projet TREC

Rapport de recherche n° 4154 —Avril 2001 —24 pages

Abstract: We study the interaction of a large number of connections over the Internet, sharing the same local loop and controlled by TCP. We focus on the case when the connections are homogeneous and the access router has a Fair-Queueing scheduling discipline, whereas the Internet is represented by a set of First-In-First-Out (FIFO) routers. We use both simulation and analytical results to determine the characteristics of the throughput obtained by each connection under a realistic description of the traffic offered to the network. The key idea for the analysis consists in a fixed point method which is based on the exact description of one TCP connection and a simplified description of the interaction with the other connections. The convergence of this fixed point method is substantiated by extensive simulations. The validation of the approach is carried out using NS simulation in cases where both methods can be used. We show that the goodput obtained by each connection and its fluctuations can be accurately evaluated from this approach.

Key-words: TCP, access router, HTTP traffic, internet, throughput, quality of service.

[*] INRIA & ENS, ENS, Département d'Informatique, 45 rue d'Ulm 75005, Paris, France
[†] INRIA & ENS, ENS, Département d'Informatique, 45 rue d'Ulm 75005, Paris, France
[‡] IBM, T.J. Watson Center, 30 Saw Mill River Road, Hawthorne, NY 10532, USA

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11 —Fax : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Méthodes de Point Fixe pour la Simulation du Partage d'une Boucle Locale par un Grand Nombre de Connexions TCP en Interaction

Résumé : Nous étudions l'interaction d'un grand nombre de connexions sur l'Internet, partageant une même boucle locale et toutes contrôlées par TCP. Nous nous concentrons sur le cas où les connexions sont homogènes et le routeur d'accès a une discipline de service du type *fair queueing*. L'Internet est représenté par un ensemble de routeurs PAPS. Nous utilisons un mélange de méthodes analytiques et de simulation pour déterminer les caractéristiques du débit obtenu par chaque connexion sous des hypothèses réalistes concernant le trafic offert au réseau. L'idée principale réside dans une méthode de point fixe fondée sur la description exacte d'une connexion contrôlée par TCP et sur une description simplifiée de son interaction avec les autres connexions. La convergence de cette méthode de point fixe est testée au moyen de nombreuses simulations. Les résultats sont aussi validés par comparaison avec des simulations en NS. Nous montrons que le débit obtenu par chaque connexion, sa moyenne et les fluctuations autour de cette moyenne, peuvent être prédits avec précision par cette approche.

Mots-clés : TCP, routeur d'accès, Trafic HTTP, Internet, débit, qualité de service.

1 Introduction

Performance characterization of the Transport Control Protocol (TCP) has been receiving increasing interest in these last years, mainly due to its dominance in the Internet. Most studies have been focused on a single TCP connection, using a variety of approaches: deterministic analysis of the steady state [9, 16], stochastic analysis of the steady state [20], fluid queueing model [7], algebraic computation [5], as well as some refined models of losses [17, 1, 2].

As it is well-known, TCP reajusts its send rate through the feedback information on the network congestion. The interaction among TCP connections is thus a crucial and difficult issue. Gibbens and Kelly [11] analyzed the bandwidth sharing of different TCP connections using a fluid model where queueing is neglected. This work was followed by other contributions, in particular that of [15].

Such bandwidth-sharing analyses can be seen as first-order characterizations of the interaction between TCP connections. While they are extremely useful, they can hardly provide information pertaining to higher order statistics which would allow one to quantify quality-of-service. As an example, when the TCP connections are homogeneous (i.e. they have the same routes and similar router characteristics, and they have similar sender and receiver buffering capacities), the bandwidth is equally shared. If timeout events are rare, and if losses are not synchronized, the bandwidth share of any TCP connection is simply the bottleneck router capacity divided by the number of TCP connections under consideration. However, the higher order statistics such as the tail distribution of instantaneous throughput depend very much on the network architecture and configuration parameters including scheduling disciplines in the routers, TCP configuration parameters, number and propagation delays of shared/unshared routers.

Another drawback of the above mentioned pieces of work is that they are only concerned with “long lived” (or infinite) traffic sources, i.e. the TCP connection always has something to send. Such sources are typical of long FTP connections. However, it is well-known that the majority of the TCP connections are those of HTTP. Most of the HTTP transfers are quite short (the so called web mice), with an average of the order of tens of Kilo Bytes [4]. Thus, these transfers use more the TCP slow-start phase than the congestion-avoidance phase, which has been the focus of the previous works.

In this paper, we propose a new simulation tool for the performance prediction of interacting TCP connections. Our technique allows us to analyze not only a large number (e.g. tens of thousands) of TCP connections competing for bandwidth sharing, but also general HTTP traffic (instead of infinite sources alone).

Simulating tens of thousands of interacting TCP connections is computationally intensive with currently available simulation tools based on the exact evolution of each flow. It is especially so when the connections are using a large number of routers.

The key idea that our simulation method is based upon is a fixed-point method that allows one to replace the exact simulation of a very large number of interacting TCP-controlled connections (which is practically speaking not feasible over a large network at this stage) by the exact simulation of one *reference connection* and a simplified description of its interaction with the other connections. In this work, we use the max-plus description of TCP proposed in [5] for the exact simulation of the reference connection. This approach, referred to as FP (fixed point) simulation, allows for a detailed description of the access router, of the routers of the IP backbone, and of the end to end control imposed by TCP.

This approach also allows for a detailed description of traffic models. In the current paper we consider Web traffic. More precisely we use a session-level HTTP traffic model [14]. We construct stochastic processes to represent the traffic offered to the network. These stochastic processes have statistical properties which are compatible with those identified by detailed statistical analysis of Web traffic in [8] and [14].

In the present paper, we shall mainly focus on the Internet as seen from a given local loop. We consider the case where a large number of TCP connections share the access router and link of this local loop. We shall focus on the case where the scheduling of the local loop router is Fair Queueing (FQ). The method can nevertheless be extended to the FIFO case as indicated in the conclusion.

Our tool provides not only detailed results on the goodput or the send rate of a given source, but also statistics on the fluctuations of these quantities over time, which is a key feature for the prediction of QoS

measures such as the tail distribution of the instantaneous throughput. In this sense, this simulator is a direct analogon of Erlang's formula for such networks in that it allows one to predict the QoS offered to a customer from statistical data on the traffic, at least under the assumptions which were described above. This allows us to analyze the total number of users that a given local loop could accommodate so as to preserve a predefined objective, say a given statistical guarantee of quality of service.

The the paper is organized as follows. In the first sections, we give a brief description of the network and traffic model, of the protocol parameters and finally of the simulator. The validation of this approach against NS simulation is presented in §6. This approach could in principle be used for other types of traffic, like video streaming or IP voice, or for mixtures of such traffic, and it could also be extended to the non homogeneous call, all via the analysis of multidimensional fixed points. This will be the object of future research.

2 Network model

The reference flow consists of a single source (a distant server) sending packets to a single destination (some user located in the local loop of interest here) over a path made of K routers in series. The transmission of packets of this reference flow is assumed to be TCP controlled. Each router is represented by a single server queue. Each queue serves the packets of the reference flow as well as those of other flows, which will be referred to as *cross traffic* flows in what follows. On the access router associated with the local loop of interest here, we assume that all cross traffic flows are statistically identical to the reference flow.

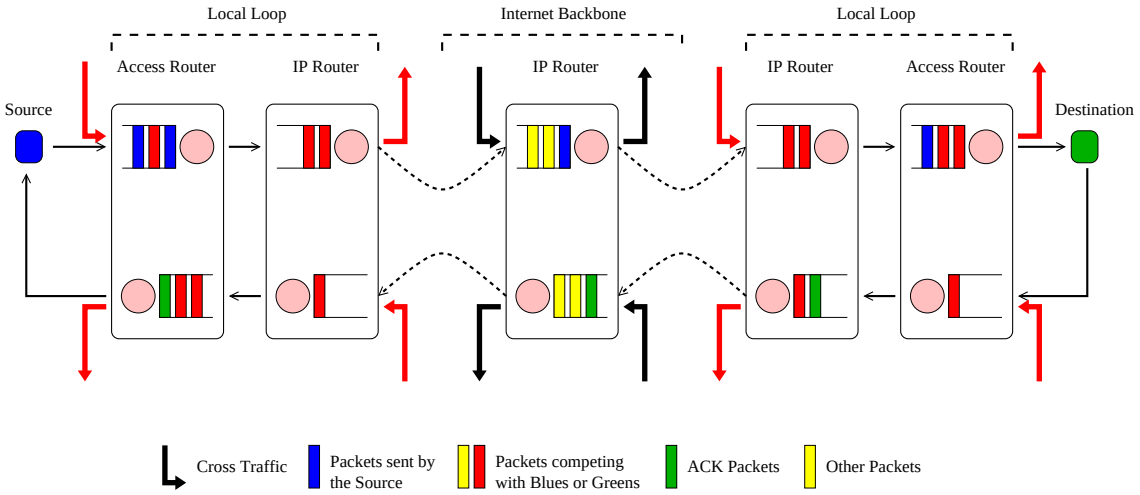

Figure 1: Connection Controlled by TCP

As shown in Figure 1, the network is composed of two types of links and routers:

- the local loops, located at the two extreme points of the reference connection; this includes all network elements connecting the end-user to the first backbone router. The local link can be established using different technologies such as cable modems, broadband satellite, ADSL, radio signals etc.,
- the backbone architecture, which connects local loops via links and routers.

A network model is then characterized by the number of routers used by the reference connection, together with the following parameters for each router:

- the scheduling policy (we will focus on FQ, but FIFO could be considered too),
- the service capacity C_s,

- the buffer capacity C_b,
- the aggregated service times (or cross traffic characteristics) – see below,
- the propagation delays between routers.

The interference of the return path (for acknowledgement packets from the user to the distant server) with the forward path (for data packets) will be assumed to be negligible.

3 Traffic model

In the following we will mainly concentrate on HTTP traffic. The traffic model that we proposed stems from the measurements and observations made in [8] and [14]. We will describe a HTTP source via its *potential traffic*, which is defined by two random sequences (cf. Fig. 2):

- the sequence $\{B_n\}_{n>0}$ describing the sizes (in number of packets) of the documents successively downloaded by the user, which will also be referred to as the *burst size* sequence;
- the sequence $\{I_n\}_{n>0}$ of think times, which represent the durations separating the end of a download from the beginning of the next one. More precisely, a download starts at the end of a think time, and each completed download is followed by a think time. In the following, the choice of the distributions of B_n and I_n has been guided by the statistical analysis of the current IP traffic using the INRIA benchmark tool WAGON [14]. These distributions have been identified as heavy tailed. A typical example would be that of lognormal distributions, which are fully determined by their mean values ($\overline{B}$, $\overline{I}$) and their standard deviations ($\widetilde{B}$, $\widetilde{I}$).

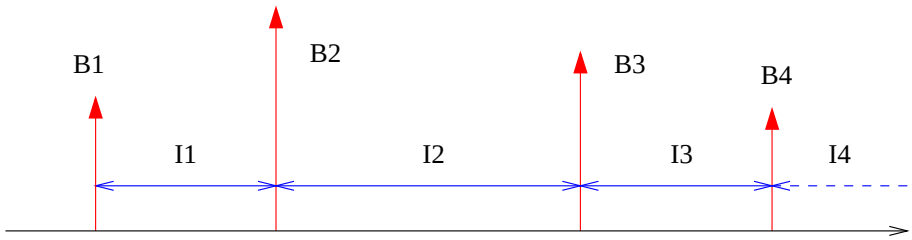

Fig. 2. Potential traffi c

In real traffic, these two sequences influence each other and may be correlated. In particular, it would make sense to assume that the user behavior is influenced by the effective throughput one obtains. Nevertheless, the simulation results which are described below are all based on the simplifying assumption that these two sequences are independent and that each of them is a sequence of i.i.d. (independent and identically distributed) random variables.

For a given network configuration, a potential traffic generates a real traffic (cf. Fig. 3) which is described in natural terms via a succession of

- off-periods (T_{off}), when the user reads the downloaded documents (think times);
- on-periods (T_{on}), when the user tries to download files.

If some user clicks to download a file of size B, the duration of the on-period is a function of the throughput (or more precisely the goodput) that this user obtains from the network at the time of the click. In particular, when

the number of users grows large, this throughput goes down due to the sharing of the capacity by TCP. This results in a stretching of the duration of the downloading of a given file, which in turn stretches the duration between the download of files, at least in the particular case when the think times are not affected by this overall slow down (throughout the paper, we will assume that the off-periods are i.i.d. with the same distribution as the think times, and that they do not depend on *on* periods). This model is close to the finite population source model of queueing theory.

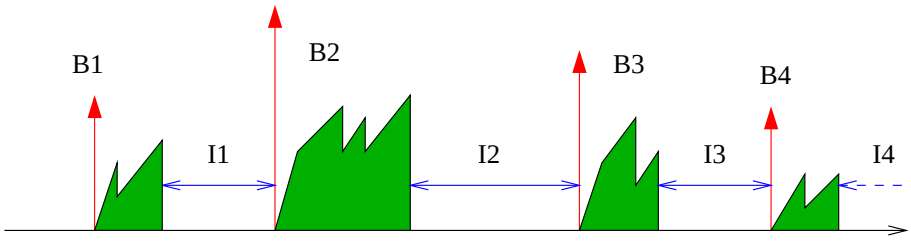

Fig. 3. Generated traffi c

4 Protocol model

The adaptive window flow control mechanism of TCP is based on the Additive Increase and Multiplicative Decrease (AIMD) principle. For a detailed description of variants of TCP protocols, the reader can refer to [21, 3].

The protocol is characterized by the following parameters:

- maximum window size ($W_{\max}$),
- maximum segment size (MSS), ack segment size,
- loss detection policy,
- timeouts detection policy (parameters for RTO, e.g. $RTO_{\min}$, update algorithm of $srtt$, $rttvar$),
- TCPtick (timer granularity),
- idleness detection,
- window adaptation algorithm to loss detection,
- window adaptation algorithm to timeouts.

In the current version of our simulator, the loss detection is assumed to be instantaneous.

5 Simulator description

5.1 Max-plus equations

We recall that the path from the source to the destination is made of K routers. We give below the definition of the key variables of the (max plus) simulator defined in [5] which is used for the fast simulation of the evolution of the reference flow:

- $y_i(k)$ is the date at which packet k leaves router i.

- $\sigma_i(k)$ is the time necessary to complete the processing of packet k measured from the epoch when this packet is *head of the line* in router i (this will be referred to as the k-th *aggregated service time* on router i – see below).
- $d_{i-1,i}$ is the propagation time from router $i-1$ to router i ($d_{K,0}$ for the way back).
- v_k is the window size at send time of packet k.
- $T(k)$ is the date when the reference source is ready to send packet k.

The following max-plus (MP for short) equations are used for the reference session:

$$y_0(k) = [y_K(k - v_{k-1}) + d_{K,0}] \vee T(k) + \sigma_0(k),$$
$$y_i(k) = [(y_{i-1}(k) + d_{i-1,i}) \vee y_i(k-1)] + \sigma_i(k),\ i = 1, .., K.$$

These equations simply translate the various queueing and window flow control constraints: for instance $y_{i-1}(k) + d_{i-1,i}$ is the arrival time of the k-th packet in router i, whereas $y_i(k-1)$ is the departure time of the $k-1$-st packet from this router; queueing on router i implies that the k-th packet becomes head of the line in its flow (i.e. within the set of packets of the referrence connection) at the latest of these two dates. The interpretation of the first equation (which translates window flow control) is similar. The value of v_k is computed adaptively according to TCP principles (see [5]).

5.2 Scheduler

As already mentioned, we focused on the case when the access router uses a FQ scheduling policy per flow. In this case, the *aggregated service time* $\sigma(n)$ of packet n of the reference flow, is the time necessary to complete the processing of packet n measured from the epoch when this packet is head of the line in its own flow in the access router.

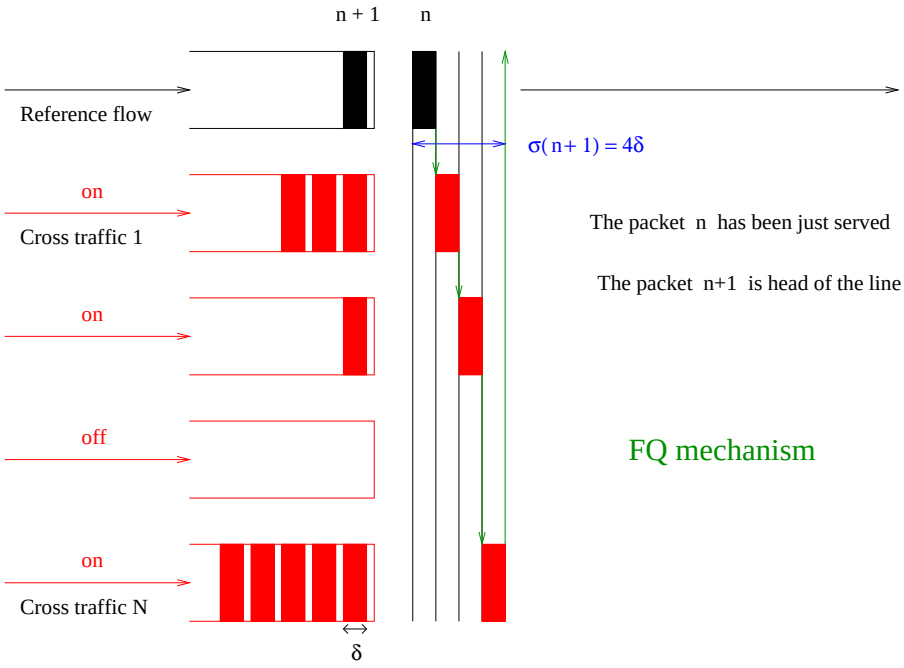

Fig. 4. FQ scheduler

For instance, assume that Figure 4 gives a snapshot of the state of the buffer of some FQ access router just after the departure of packet n of the reference flow. When assuming that the server visits sequentially

the queues of the flows in the order indicated on the figure, and that no further arrivals take place on the cross traffic flows until the server returns to the reference flow, since there are 3 cross traffic packets to serve before the server can return to flow 1, $\sigma(n+1)$ is then the time necessary to serve 4 packets (3 cross traffic packets plus one reference flow packet). This quantity, which can be built during the execution of the FQ simulation procedure, is then used in the max-plus product iteration representing the evolution of the reference flow as indicated in the above equations.

5.3 Fixed Point

We now describe the fixed point algorithm in the simplest possible case, which we call the basic local loop model (see Fig. 5 below): N statistically independent HTTP flows compete for the bandwidth of the access router and all other routers are assumed to be "infinitely" fast, so that no queueing delays are incurred there (i.e. only propagation delays are taken into account in the backbone). This basic model will be enriched to a network with queueing delays as described in Figure 1 in the following sections.

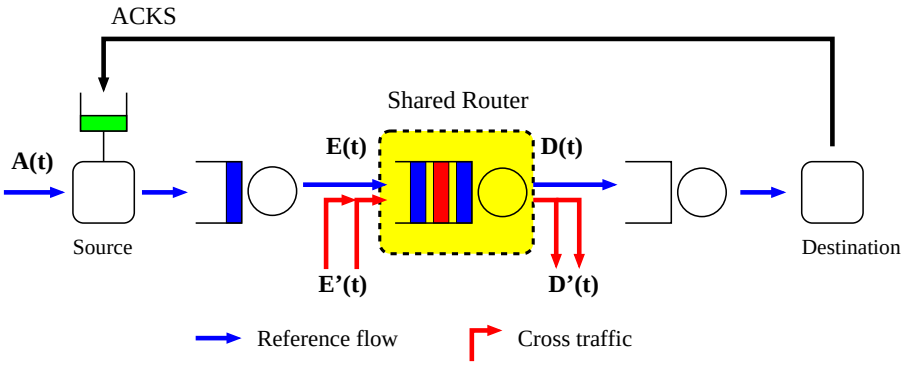

Fig. 5. Cross traffi c on shared router

In order to construct the stationary regime of the dynamics of the reference flow, we use *fixed point schemes*, which are the main new idea (in comparison to [5]). The general idea is the following:

- If aggregated service times were known, then one could use the MP equation to recursively compute the departure (resp. arrival) times of all packets of the referrence flow. This would in particular determine the point process of the arrivals to the shared router and/or the long time average of the throughput for the reference connection; by symmetry, we would then also know these characteristics for each connection.
- Similarly, if one would know the arrival point process and/or the average throughput of each connection, then using the scheduler one could determine the sequence of aggregared service times of the reference connection.

So we can take one of these (either the aggregated service times or the point processes) as an unknown variable to be updated in a joint simulation of the MP equations and the scheduler, which is run until a fixed point is reached.

Scheme 0 below, which is the simplest one, is only presented for the sake of clear exposition since we shall see that it lacks sufficient symmetry to work. Scheme 1 is the simplest scheme that leads to satisfactory results. Both scheme 0 and 1 take as fixed point variable the intensity (long term average) of a connection. Scheme 2, which can be seen as a refinement of Scheme 1, takes the aggregated service times as fixed point variables. Scheme 2 requires more simulation time, but allows one to take into account various fine phenomena such as flow synchronization, which Scheme 1 cannot capture.

Fixed point on the point processes (Scheme 0)
For all $0 < \gamma < \infty$, where γ represents the send rate of a source, measured in number of packets per second, and conditionaly on the fact that this source is *on*, and for all sequences $\{(I_n, B_n)\}$ which have the characteristics described above for the potential traffic of a session, let $A(\gamma)$ be the point process which alternates between i.i.d. *off* phases and i.i.d *on* phases as follows:

- The n-th *off* phase has length I_n;
- The n-th *on* phase consists in the good transmission of B_n packets. The interarrival between two packets has some distribution function with distribution function F_γ on the real line in some parametric class characterized by its mean γ^{-1}.

Scheme 0 consists in the following fixed point scheme on the point processes of this class:

Step 1:

1. Generate N i.i.d. copies of $A(\gamma_0)$, with $\gamma_0 = \frac{C_*}{N}$.
2. The generated point processes are injected into the shared router to calculate the sequence $\{\sigma^{(1)}(n)\}$ of aggregated service delays at this router.
3. Then, by max-plus products iteration, one gets the averaged throughput γ_1 obtained by the reference connection when this one is on.

Step 2:

1. Generate N i.i.d. copies of $A(\gamma_1)$.
2. The generated point processes are injected into the shared router to get the sequence $\{\sigma^{(2)}(n)\}$.
3. By max-plus products iteration, one gets γ_2.

And so on.

The stationary throughput is then given by the limit $\gamma_\infty = \lim_{k\to+\infty} \gamma_k$, when it exists, and the stationary transmission time sequence by the sequence of transmission times of the reference flow obtained when all cross traffics are of the form $A(\gamma_\infty)$.

The rationale for Scheme 0 is twofold:

1. the symmetry, or equivalently the fairness of TCP in this FQ and homogeneous setting, which allows one to state that when it is on, the reference flow should get a long term average throughput which is the same as that of any of the cross flows;
2. the high variability assumption on the potential traffic, which justifies the assumption that the windows of the various flows are not synchronized, and which translates here into the independence assumptions between the N cross flows.

Here are two natural ways of improving this scheme:

- the copies of the reference flow used to construct the cross traffic flows are based on a parametric model of the inter-arrival times characterized by its k first moments and on a dynamic estimation of these first moments from the measurements on the reference flow;
- the dynamic estimation of these first moments is made conditionally on the current number of active flows sharing the router. For instance, conditionally on the current value of the number of *on* sources, say n, the packets of each of the cross-traffics arrive with interarrival times sampled as an i.i.d. sequence with distribution $F_n(x)$ as obtained from the measurements on the reference flow etc.

Nevertheless, even with the last improvements, Scheme 0 has the following drawback: the cross flows send at a rate which is the same as that obtained by the TCP controlled reference flow, like in TFRC [10]. But this is not a sufficient symmetry (the reference flow is controlled by an adaptive window size mechanism whereas the cross flows are contolled by their rate only), and thus no fair fixed point can be reached in general (a fixed point is fair if the mean values of throughput and of queue size are the same for the reference connection and each cross flow).

The following schemes (1 and 2) were developped to enforce more symmetry and in particular to take into account the fact that in the shared router, the number of packets of each cross flow should be the same in law as that of the reference flow due to symmetry, a property that the equality of rates is not sufficient to yield.

Shared router queue size constraint on cross traffic flows (Scheme 1)
Scheme 1 consists in adding to Scheme 0 a mimicking of the effect of the window flow control on each cross flow based on the following: the arrivals of each cross flow on the shared router are delayed so as to maintain the shared router queue size of this flow equal in distribution to the shared router queue size of the reference flow.
Scheme 1 and its conditional version (see above) allow one to reach fair fixed points in cases where synchronization is limited. However, if one suspects that most flows will experience synchronous losses, one should rather use Scheme 2.

Window flow on cross traffic flows (Scheme 2)
In scheme 2 the fixed point now bears on the aggregated service time distribution G, rather than on the interarrival time distribution function as above. In addition, a more realistic window flow control is implemented on each cross flow, via a simplified (max,plus) representation, with current window size $w(n)$ for cross flow n;

Step 1:

1. At each arrival time of cross flow n, the next arrival time of this flow is generated from the knowledge of the current value of the window size $w(n)$ and from the fact that the packet in question is lost or not; the value of this arrival time is computed via a simplified (max,plus) recurrence associated with this flow, which requires that the w^* last arrival times of this flow be kept in memory, and also that a new aggregated service time with law $G_0 = G$ be sampled. Timeouts are also taken into account as occuring with frequency p_0.
2. Using these arrival processes in the scheduler, one builds the sequence $\{\sigma^{(1)}(n)\}$ of aggregated service times of the reference flow, with steady state distribution G_1. This is used in the (max,plus) equations of the reference connection to analyze the frequency of timeouts, p_1.

Step 2:

1. At some arrival time of cross flow n, the next arrival time of this flow generated as above but with a new aggregated service time with stationary law G_1, and with timeouts occurring with frequency p_1.
2. This gives the sequence $\{\sigma^{(2)}(n)\}$ of aggregated service times on the shared router for the reference flow. with steady state distribution G_2.

And so on.

The laws G_i are again based on a parametric model characterized by its first k moments and on a dynamic estimation of these first moments from the measurements on the reference flow.

A natural improvement consists in making the dynamic estimation of these first moments conditionally on the current number of active flows sharing the router.

5.4 Convergence tests

Attention should be paid to the fact that in these iterations, there is no theoretical guarantee that the fixed point exists and that the procedure converges to such a fixed point. Nevertheless, we never found cases where the procedure associated with Scheme 1 or 2 did not converge to a fair fixed point.

Here are a few hints for checking convergence to the desired fixed point by simulation:

1. *Environment convergence*. We have to check the statistical convergence of input variables (burst size, think time). The number of iterations should be large enough to guarantee that in Cesaro mean, these variables are close to their means. By the central limit theorem, it is easy to notice that for i.i.d. random variables ξ_i with mean M equal to their standard deviation and equal to 50, the deviation $|\sum_{i=1}^{N} \frac{\xi_i}{N} - M|$ is smaller than 1% of M with probability higher than 0.95 if $N > 40000$. With an iteration of 10^6 packets of the reference flow, there are about 20000 bursts simulated, leading approximately to 1.4% of deviation. This also takes into account the stabilization of the generated environment, like loss or timeout probabilities, etc.

2. *Fixed point convergence*. The fixed point constructed above must satisfy several constraints. In particular the statistical characteristics (on-period, buffer size by flow at the shared router, etc.) of the reference flow and of each cross traffic flow are to be close enough.

The above requirements and checks are integrated in the current version of the simulator.

6 Comparison to NS

In order to check the validity of our fixed-point simulation method, we carried out simulations using NS and compared the results obtained using the two methods.

The typical network configuration we simulate is composed of 5 network elements: Source (**0**) - Access Router (**1**) - IP router (**2**) - Access Router (**3**) - Destination (**4**). The access router (3) will play the role of the shared router. We assume that the return route for acks is separated from the forward route. This backward route is represented by a constant propagation delay from the destination to the source.

In these simulations there are several major constraints which limit the validation scheme:

- for NS, the number of parallel sessions should not be too big in order to keep acceptable simulation times;
- for the FP simulator, the number of parallel sessions should not be too small, otherwise the mean-field argument may be meaningless;
- the NS simulations allows for FIFO with shared buffer and for FQ with individual buffers (one buffer for each flow);
- the FP simulations are implemented for FQ with shared and individual buffers.

We simulate the case of N FTP sources, which is the case where one can expect the discrepancy between the two methods to be the largest as losses have more chance to be synchronized in this case than in the HTTP case.

The simulated configurations have the following characteristics: $N = 30$, $C_s(\text{shared router}) = 125$ pkts/s, $MSS = 8$ Kb, $RTT_{\min} = 140$ ms, $W_{\max} = 20$ pkts, $TCPtick = 100$ ms, $RTO_{\min} = 2 \times TCPtick$. All routers are with FIFO policy, except for the access routers which may be FQ with shared buffers or FQ with individual buffers.

Results for the case of individual buffers

- **NSWI**: NS simulation with FQ individual buffers (5, 10 pkts);
- **FPWI**: FP simulation with FQ individual buffers (5, 10 pkts).

Table 1.

	NSWI-5	FPWI-5	NSWI-10	FPWI-10
p_{loss}	4.00 %	4.54 %	1.45 %	1.59 %
p_{to}	0.00 %	0.00 %	0.00 %	0.00 %
RTT(s)	0.96	0.85	1.71	1.69
λ_g(pkt/s)	4.16	4.16	4.16	4.16
λ_s(pkt/s)	4.34	4.29	4.23	4.21

p_{loss}: loss probability, p_{to}: timeouts probability, RTT: mean round trip time, λ_g: goodput, λ_s: send rate.

For this FQ case with individual buffers, the general agreement is rather good (cf. Table 1). Discrepancies are probably explained by inherent differences between the two simulators: losses and timeouts are detected instantaneously in the FP case, and with delay in the NS case. There is more synchronization of sources in the NS case etc.

Results for the case of shared buffers For reasons explained above, for the shared buffer FQ case, we could not compare the results obtained by NS and our method due to the lack of implementation of this discipline within NS. In place, we considered the following two cases:

- **NSFS**: NS simulation with FIFO shared buffer (150, 300 pkts);
- **FPWS**: max-plus simulation with FQ shared buffer (150, 300 pkts);

Of course, there is no reason for these two to coincide, although one can notice that in this particular case, the results are still rather close (cf. Table 2).

Table 2.

	NSFS-150	FPWS-150	NSFS-300	FPWS-300
p_{loss}	3.91 %	3.77 %	1.39 %	1.85 %
p_{to}	0.00 %	0.00 %	0.00 %	0.00 %
RTT(s)	1.08	1.29	1.90	2.07
λ_g(pkt/s)	4.16	4.16	4.16	4.16
λ_s(pkt/s)	4.31	4.33	4.22	4.24

Note that in all these FTP examples, the average goodput $\overline{\lambda}_g$ is exactly the ratio C_s/N. This can be explained by several facts: losses are not too much synchronized (very synchronized losses and small buffers naturally lead to an underutilization of the bandwidth); in addition, there are no timeouts, so that all the load brought to the access router is good load.

These examples and others show that our approach leads to results which are reasonably close to NS simulation results at least for the considered types of traffic assumptions. An extensive simulation is to be done in the future for a more complete validation.

7 Simulation of a large number of sources

7.1 Local loop studies

In the simulations below, we have taken the following parameters:

- Protocol parameters:
  - maximum window size: 40 pkts.
  - MSS: 1.5 Kbytes, ack: 0.04 Kbytes.

  - loss detection policy: instantaneous.
  - $TCPtick = 0.5$ s.
  - $RTO_{\min} = 2 \times TCPtick$.
  - idleness is detected after 10 s. When an idlenss of more than 10s occurs, a new TCP session is established starting with slow start.
  - algorithm in case of loss detection: Reno-type ($W_{n+1} := W_n/2$).
  - algorithm in case of timeouts: restart with $W_{n+1} := 1$.

- Network parameters:
  - Source: $C_s = 10$ Mb/s, $C_b = \infty$.
  - Access Router: FIFO, $C_s = 100$ Mb/s, $C_b = 10000$ pkts.
  - IP router: FIFO, $C_s = 1$ Gb/s, $C_b = 10000$ pkts.
  - Access router (shared router): FQ, $C_s = 100$ Mb/s, $C_b = 10000$ pkts.
  - Destination: $C_s = 10$ Mb/s, $C_b = \infty$.
  - Propagation delays: $d_{0,1} = d_{1,2} = d_{2,3} = d_{3,4} = 10$ ms, $d_{4,0} = 60$ ms.
    $\Rightarrow RTT_{\min} \simeq 100$ ms.

- Potential traffic:
  - Burst size B_n: i.i.d. lognormal with $\overline{B} = 50$ pkts, $\widetilde{B} = 50$ pkts.
  - Think time I_n: i.i.d. lognormal with $\overline{I} = 30$s, $\widetilde{I} = 30$s.

In this case, the interaction of different flows is only located at the access router. There is no cross traffic competing with the reference flow at other routers (cf. Fig. 6).

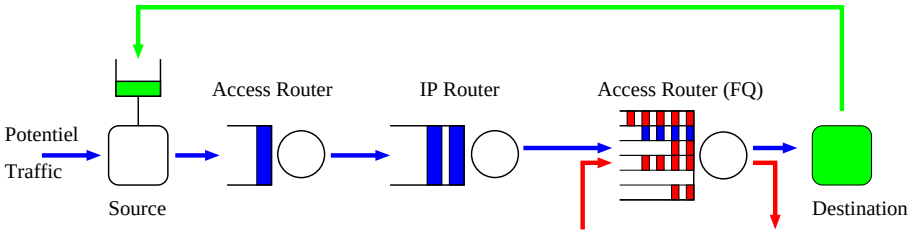

Fig. 6. Local loop model

Figure 7 illustrates the send times of the reference flow under the above configuration with $N = 7500$.

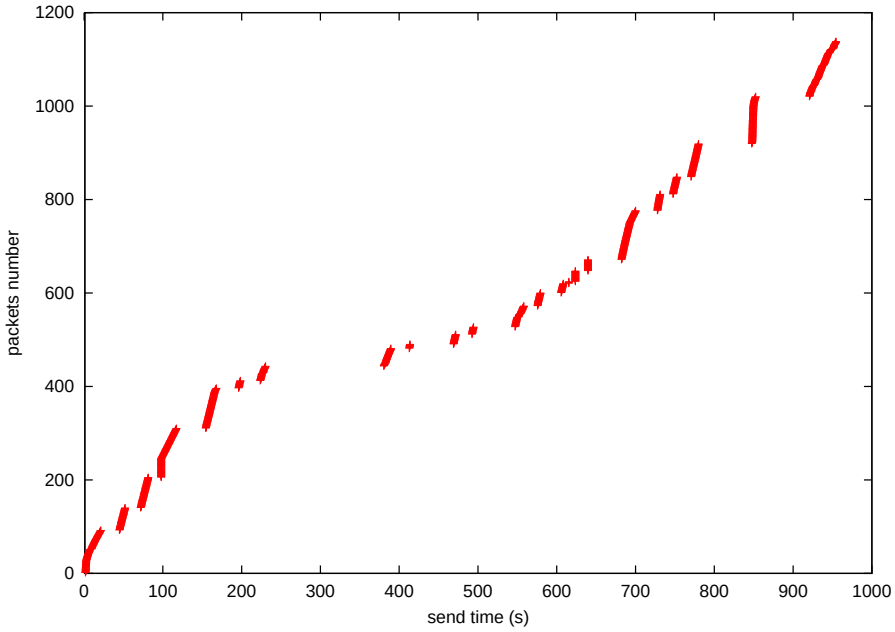

Fig. 7. Send time of packets

Table 3 summarizes various QoS values obtained after the simulation of 10^6 packets of the reference flow; here

$$\lambda_{x\%} = \max_y \{\mathbb{P}(B_n/(T_{on})_n \geq y) > x/100\}$$

is the largest instantaneous goodput that one can guarantee to each user at least $x\%$ of the time; by definition, the instantaneous goodput is measured by burst (it is the time needed to download a typical file). In the simulation, this is estimated with a granularity of 10 Kb/s.

Table 3. Local loop

nb of sources	2500	5000	7500	10000
p_{loss}	0.00 %	0.00 %	4.19 %	7.32 %
p_{to}	0.00 %	0.00 %	0.16 %	0.60 %
RTT (s)	0.10	0.21	1.77	2.21
λ_g (Kb/s)	1151.9	775.2	41.9	20.6
λ_s (Kb/s)	1151.9	775.2	43.8	22.4
$\lambda_{95\%}$ (Kb/s)	350	300	40	20
$\lambda_{90\%}$ (Kb/s)	440	370	40	20
$\lambda_{80\%}$ (Kb/s)	530	450	50	20
$\lambda_{50\%}$ (Kb/s)	970	690	50	30

Figure 8 gives the empirical distribution of the instantaneous throughput obtained for $N = 5000$.

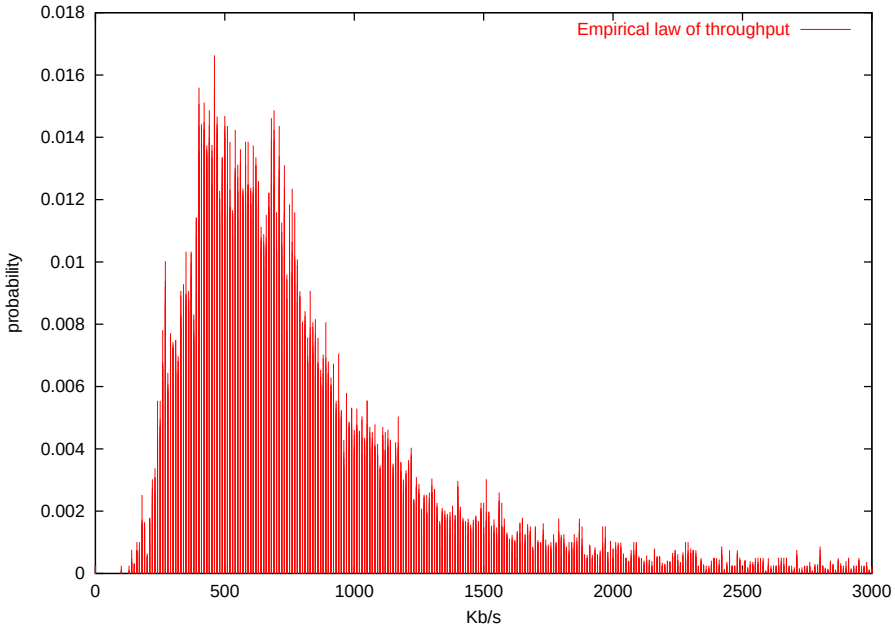

Fig. 8. Empirical distribution

7.2 Cross traffi c in the IP Backbone routers

The previous model was based on the assumption that the shared router (access router) was the bottleneck for the reference flow. In this section, we focus on the case when one of the Backbone routers may also be a bottleneck (cf. Fig. 9). The bottleneck router of the Backbone is represented as a FIFO router and the cross traffic on this router is represented as a fractional Brownian motion, as identified by previous studies on the matter [13, 8, 12, 18]. This self-similar traffic is characterized by its mean, its variance and its Hurst parameter. Below the mean is chosen such that the router is loaded up to 90% and 100% of its service capacity with the standard deviation equal to the half of the mean; the Hurst parameter is taken equal to 0.7. All other configuration parameters are as in the previous section, but for the service capacity of the IP router which is reduced to 80 Mb/s ($< C_s$(shared router)).

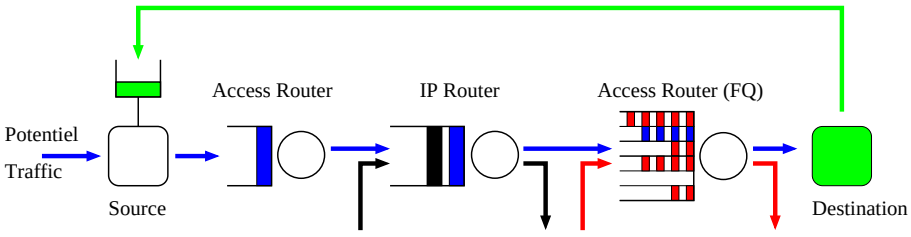

Fig. 9. Cross traffi c in Backbone IP router

Table 4 summarizes various simulated values obtained after 10^6 packets of the reference flow, with a backbone router loaded to 90%.

Table 4. 90% load

nb of sources	2500	5000	7500	10000
p_{loss}	0.00 %	0.00 %	3.90 %	7.25 %
p_{to}	0.00 %	0.00 %	0.16 %	0.60 %
RTT (s)	0.16	0.25	1.86	2.22
λ_g (Kb/s)	750.8	598.0	41.5	20.5
λ_s (Kb/s)	750.8	598.0	43.3	22.2
$\lambda_{95\%}$ (Kb/s)	190	180	40	20
$\lambda_{90\%}$ (Kb/s)	270	250	40	20
$\lambda_{80\%}$ (Kb/s)	400	360	50	20
$\lambda_{50\%}$ (Kb/s)	760	610	50	30

In Table 5, the backbone router loaded to 100%.

Table 5. 100% load

nb of sources	2500	5000	7500	10000
p_{loss}	0.20 %	0.19 %	1.81 %	5.13 %
p_{to}	0.15 %	0.17 %	0.24 %	0.68 %
RTT (s)	0.59	0.60	2.95	2.96
λ_g (Kb/s)	162.1	158.8	38.4	19.9
λ_s (Kb/s)	162.7	159.4	39.2	21.2
$\lambda_{95\%}$ (Kb/s)	30	30	20	20
$\lambda_{90\%}$ (Kb/s)	50	50	30	20
$\lambda_{80\%}$ (Kb/s)	90	90	40	20
$\lambda_{50\%}$ (Kb/s)	350	330	50	30

We will refer to the last three tables (Table 3,4 and 5) as cases 1,2 and 3 respectively. In each of these cases, the rate of the potential traffic is given by $\overline{B}/\overline{I}$, under the assumption $T_{on} << T_{off}$. From this simple remark, we see that there is a natural threshold for $N \simeq C_s(pkts/s) \times \overline{I}/\overline{B}$ parallel connections. Here this threshold is equal to 5000.

The dependence of the main quantities in N is illustrated by the 5 following set of curves (Fig. 10a-e).

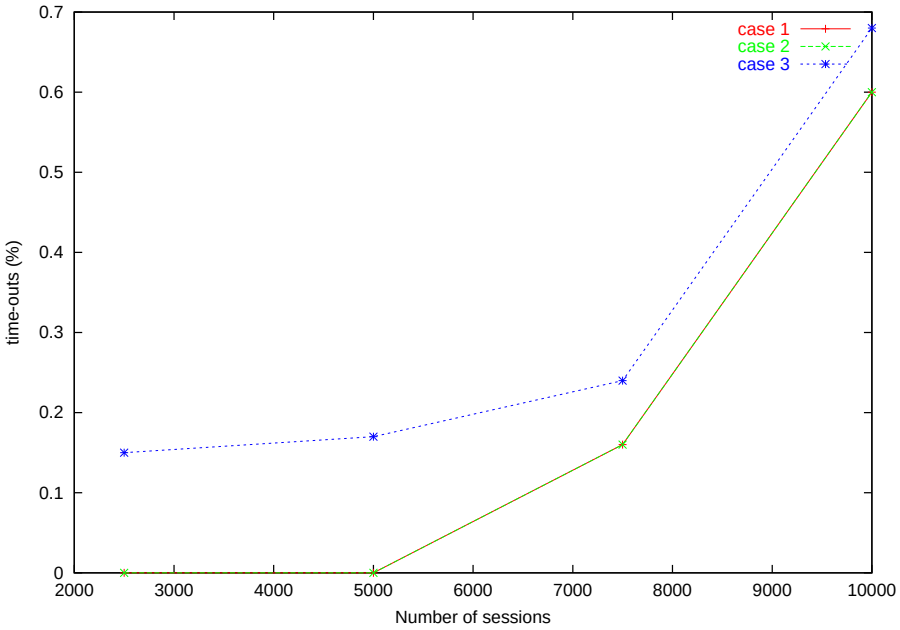

Fig. 10a. timeouts plot

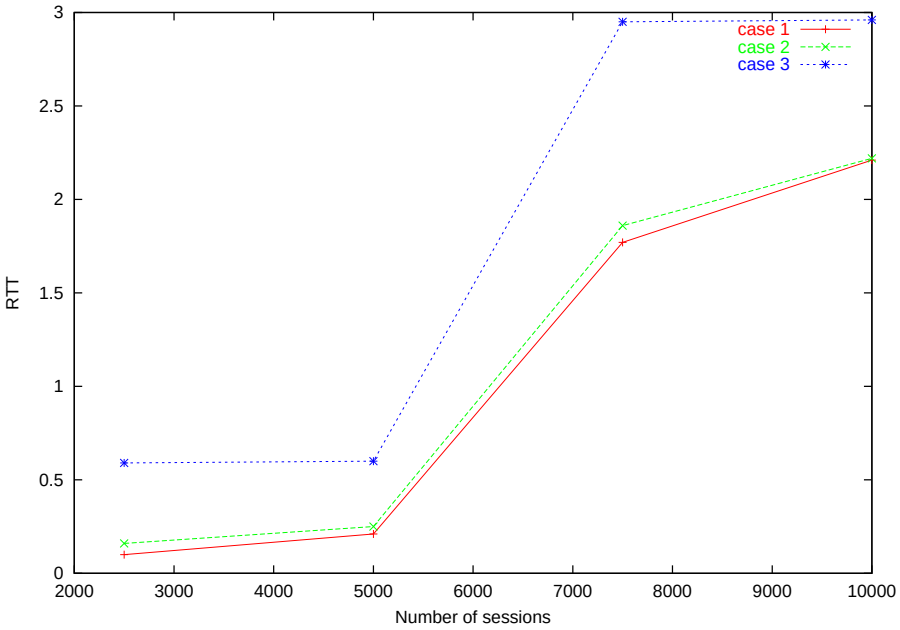

Fig. 10b. RTT plot

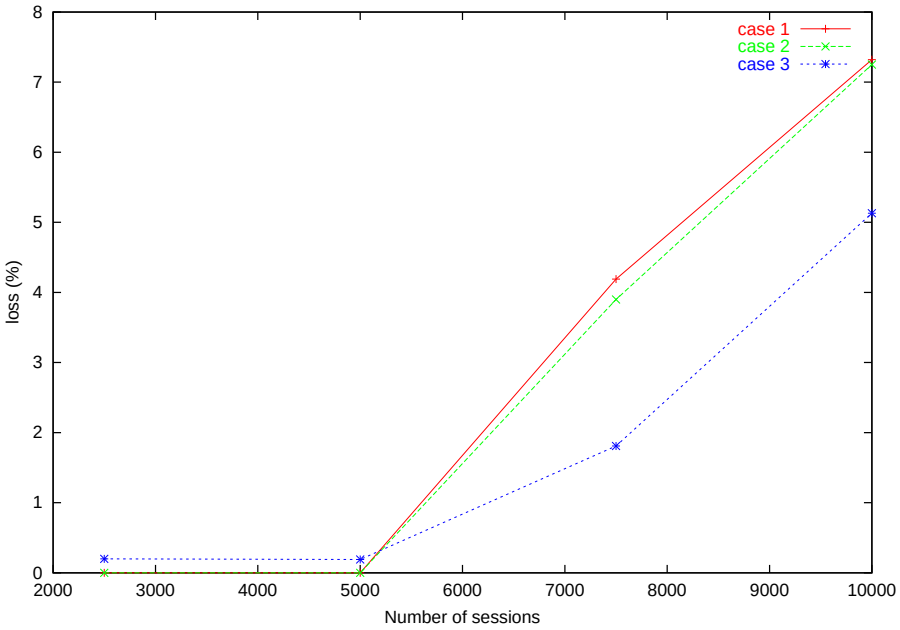

Fig. 10c. Loss plot

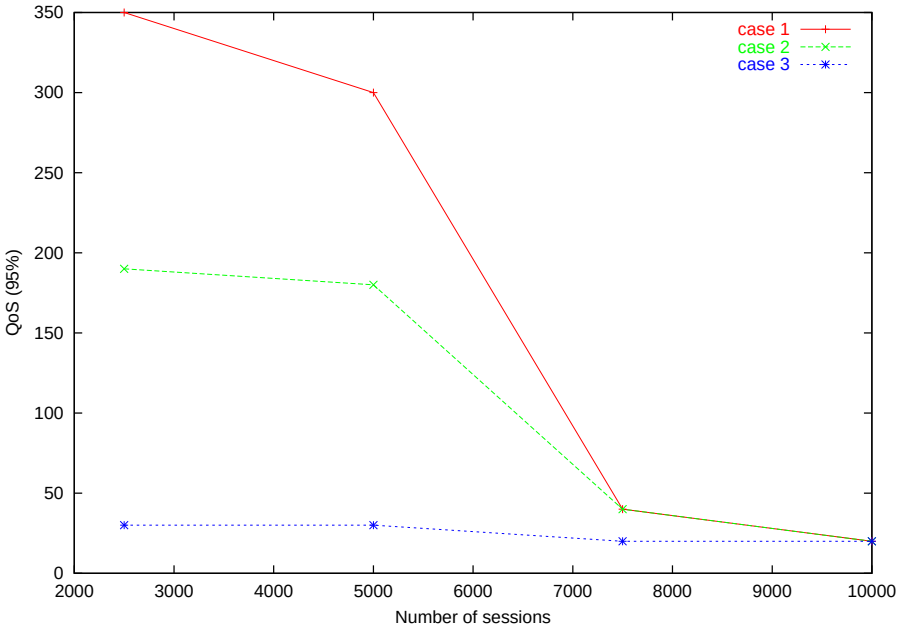

Fig. 10d. QoS plot

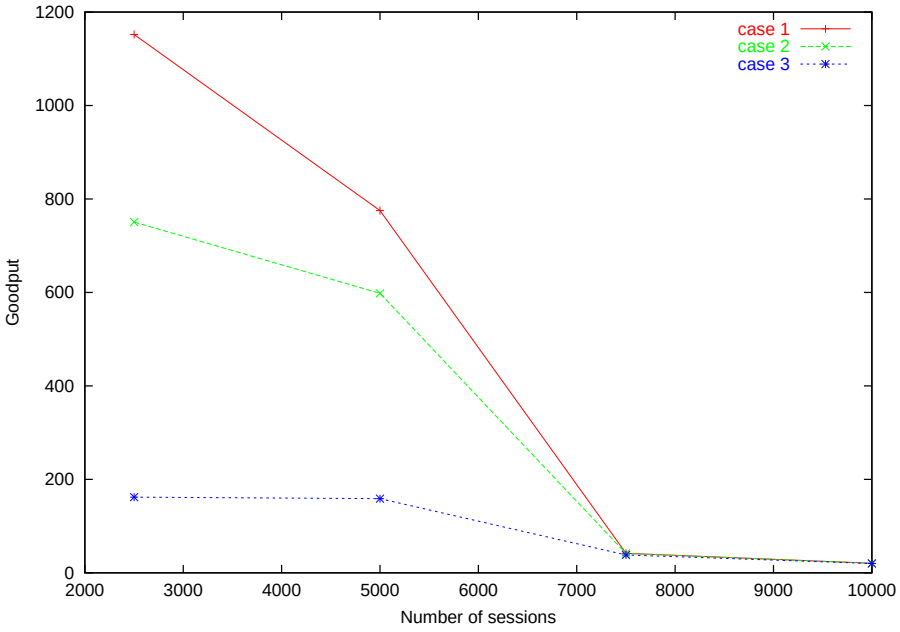

Fig. 10e. Goodput plot

- In case 1, we see that when $N \leq 5000$, no losses and no timeouts are detected: the performance degradation from $N = 2500$ to $N = 5000$ is essentially due to the increase of the RTT.
- In case 2, the bottleneck router in the backbone introduces additional delays (no losses and no timeouts) which lead to a degradation of performance. As we see, an increase of N to 7500 or even 10000 does not introduce additional losses or timeouts. On the contrary, the number of losses decreases. This may be explained by the fact that packets are less concentrated in the access router as they are more often waiting in the backbone router.
- In case 3, the bottleneck router in the backbone introduces additional delays and also additional losses and timeouts (resp. around 0.2% and 0.15%). Performances are worse than in cases 1 and 2. The impact of delay increase seems here (as well as in case 2) higher than the gain on the loss probability.

8 Fluctuations of Instantaneous Throughput and QoS Measures

The indicator of QoS which was selected in the previous section measures the fluctuations of the instantaneous goodput obtained by one user when downloading a file with the HTTP protocol.

This instantaneous goodput is a local averaging of the goodput at time t. There are several ways of making this local averaging, each with its own merits depending on the applications:

1. the averaging is made over ξ consecutive packets within an *on* period (namely when the user is continuously downloading files);
2. the averaging is made over the packets of a downloaded file (burst);
3. the averaging is made over the packets of a downloaded file, given that this number of packets is larger than ξ.

We found out that when using the first definition, the QoS functions often have steps that make it difficult to tune the system from a target QoS defined from a single value of ξ (cf. Fig. 11). A uniform averaging of these

functions over the values of ξ would be the most convenient way to define QoS in this setting. The two other definitions of QoS do not have this drawback. However, the second one is a quite stringent QoS requirement in case of high variances, where there are many small bursts, which remain most of the time in the slow start phase.

Figure 11 gives the tail probability of QoS functions of type 1 and 2 (namely the probability that the instantaneous goodput averaged over ξ packets or over the burst is more than x):

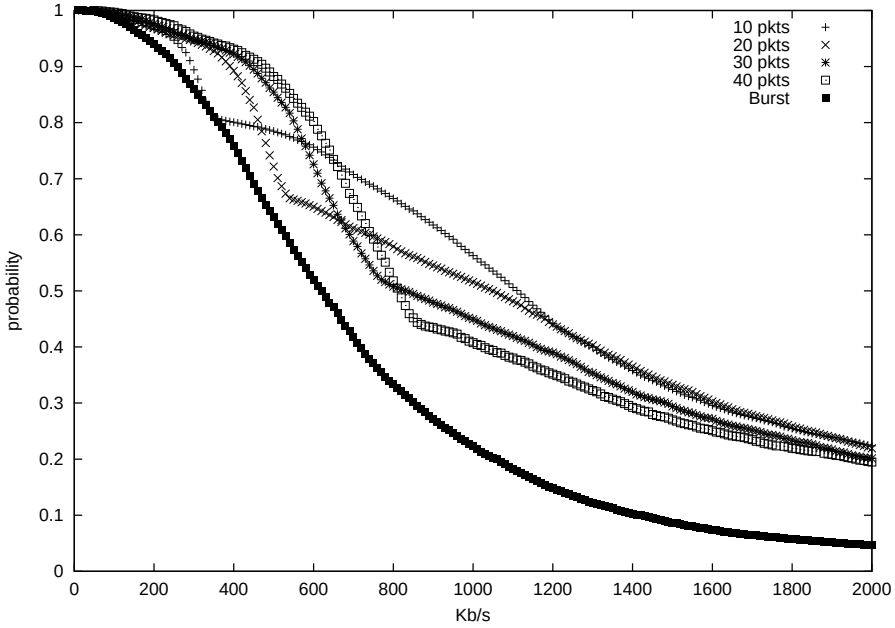

Fig. 11. Tail probability of QoS

9 Impact of Variance

The increase of variance may have a serious impact on performances, as it has been noticed in [5]. This is well illustrated by Table 6 where we give the different values in the local loop model, with all mean values fixed, but with a multiplicative scaling of the standard deviation. The case reported here is that with $N = 5000$. We will study four cases for which we will use the following notation:

- **1-1**: $\overline{B} = 50$, $\widetilde{B} = 1 \times \overline{B}$; $\overline{I} = 30$, $\widetilde{I} = 1 \times \overline{I}$.
- **5-1**: $\overline{B} = 50$, $\widetilde{B} = 5 \times \overline{B}$; $\overline{I} = 30$, $\widetilde{I} = 1 \times \overline{I}$.
- **5-5**: $\overline{B} = 50$, $\widetilde{B} = 5 \times \overline{B}$; $\overline{I} = 30$, $\widetilde{I} = 5 \times \overline{I}$.
- **10-10**: $\overline{B} = 50$, $\widetilde{B} = 10 \times \overline{B}$; $\overline{I} = 30$, $\widetilde{I} = 10 \times \overline{I}$.

Table 6.

	1-1	5-1	5-5	10-10
p_{loss}	0.00 %	0.00 %	0.03 %	0.08 %
p_{to}	0.00 %	0.00 %	0.00 %	0.01 %
RTT (s)	0.21	0.25	0.32	0.36
λ_g (Kb/s)	775.2	883.0	873.9	780.9
λ_s (Kb/s)	775.2	883.0	874.2	781.6
$\lambda_{95\%}$ (Kb/s)	300	170	200	190
$\lambda_{90\%}$ (Kb/s)	370	210	230	220
$\lambda_{80\%}$ (Kb/s)	450	250	260	250
$\lambda_{50\%}$ (Kb/s)	690	380	570	410
Idleness	1.65 %	1.65 %	0.77 %	0.65 %

In the above simulation, a change of the scaling factor of the standard deviation from 1 to 5 leads to a variation in performances of more than 10%. In this case, the scalings on the burst size or on the think time may have very different consequences. In fact, there are several mechanisms that enter into competition:

- *The pure variance impact*: the degradation of throughput when variance increases is well documented for max-plus linear systems (see [6]). This also may create losses or timeouts (case 5-5, 10-10). However, this is not the dominant mechanism between case 1-1 and 5-1.
- *The influence of* $W_{\max}$: compared to $W_{\max} = 40$, a mean burst size of 50 packets is "small", in the sense that to transmit 50 packets, one needs at most a window size of 32 ($1 + 2 + 4 + 8 + 16 + 19 = 50$), so that the full possibility of the maximum window size is not used. Increasing the variance, there are more large bursts (in case 5-1), and the maximum window size is better used: the observed mean window size indeed increases of about 7 units when moving from 1-1 to 5-1.
- *The idleness detection*: the scaling on the think time modifies the proportion of idle states. The increase of this proportion has a negative impact on the throughput, if one only considers the fact that idleness restarts the window mechanism. Clearly, this impact is not the dominant element here.

10 Comparison with Earlier Approaches

The aim of this section is to compare our results to formulas for the throughput based on the single bottleneck heuristic ([16, 19, 20]) and on the FTP source assumption. We have tested the following simple formula, naturally adapted to the HTTP source case:

$$\overline{\lambda}_s \sim \frac{1.22MSS}{RTT\sqrt{p_{to} + p_{td} + p_{idle}}}, \tag{1}$$

Table 7.

nb of sources	2500	5000	7500	10000
case 1	1112.9	538.9	39.8	23.6
case 2	702.7	464.1	39.1	23.6
case 3	195.0	188.5	34.9	20.6

This kind of formulas requires that the loss and the timeout probabilities as well as the mean RTT values be known or already estimated. In that, they are quite different in nature from the results obtained from our fixed point approach, where all these quantities are obtained from the model. In Table 7 we see that the difference with the simulated throughput may be greater than 30%. One natural explanation is that such formulas are more adapted to FTP sources where there are no fluctuation of the demand traffic. They are also known to be only applicable in case of very small loss probabilities.

The results that we obtain are also different in nature, particularly so for QoS measures, which pertain to stochastic objects. An interesting question is then to determine the range of parameters for which the mean goodput can be analyzed from considerations only based on mean value arguments. We can partly answer this question as follows. Assume that

1. the shared router behaves like a *deterministic bottleneck*; this is not necessarily the case when there are several routers which are close to bottleneck and when fluctuations are high (in which case (2) is not valid and we have to represent the throughput of each connection as a max-plus Lyapunov exponent (see e.g. [5] and the references therein);

2. the system is fair in that everybody receives in mean the same share of the total capacity C_s, which makes sense when the number of users is large, all users have homogeneous behavior, and under the assumption of a fair queueing discipline.

Then the average accepted rate λ_a is such that

$$N.p_{on}\lambda_a = C_s, \tag{2}$$

where $p_{on} = T_{on}/(T_{on} + T_{off})$. Assume in addition that there are no timeouts outside those created by losses; Then $\lambda_a = \lambda_g$ and one deduces from this that

$$\lambda_g = \frac{C_s}{N - \frac{T_{off}C_s}{\overline{B}\times MSS}}. \tag{3}$$

In this case, the goodput can indeed be evaluated directly from the only knowledge of mean values of traffic and from the total number of users. Besides, Formula (3) gives in all cases the upper-bound of the goodput.

In order to proceed further, one needs of course to have an estimate of the loss probability p_{loss}. This cannot be inferred from considerations in mean unfortunately. This is a functional of the buffer size, the statistics of burst and idle periods, the scheduling etc.

One should also be careful with the use of this approximation in the absence of a precise knowledge of the probabilities of times outs; it was reported in [19] and [20] that timeouts are in fact quite common, often an order of magnitude more than pure losses.

Let us conclude this discussion by stressing that high-order statistical QoS measures can in no case be estimated without some detailed knowledge of the statistical nature of traffic. Hence, even when formula (3) can be used to approximate goodput, the prediction of QoS cannot be derived from considerations in mean.

11 Conclusions

In this paper we have proposed a novel simulation technique for predicting the interaction of a large number of TCP connections. The model of offered traffic is that of HTTP sessions. Our tool is based upon a fixed-point method combined with mean-field type of arguments, and allows for a detailed algebraic description of the network architecture and the TCP protocol, as well as the traffic model.

The presentation has been focused on the case of homogeneous TCP connections sharing a local loop. The scheduling discipline of the access router is FQ, whereas that of the Internet backbone routers is FIFO. It is straightforward to extend the tool to describe other scheduling disciplines in the local loop as well as in the Internet backbone.

The current paper provides a tool for analyzing detailed statistics of a large number of interacting TCP connections sharing a local loop. Our on-going work includes the extension of the tool to the case of heterogeneous TCP connections. The key issue here is to design an efficient scheme for the multi-dimensional fixed point calculation.

It is worthwhile noticing that our fixed-point method can also be used in NS simulators. In such a case, only the reference flow is simulated as a pure TCP connection. The cross traffic flows are simulated using one

or more UDP connections. The implementation of such an NS simulator would require a number of auxiliary variables to describe the instantaneous performance characteristics of the reference flow.

Acknowledgment. The authors are grateful to Naceur Malouch of INRIA for carrying out NS simulations presented in the paper.

References

[1] Altman, E., Barakat, C. and Abratchenkov, K. "TCP in presence of bursty losses". *Performance Evaluation*, Vol. 42, Issues 2-3, pp. 129-147, 2000.

[2] Altman, E., Barakat, C. and Abratchenkov, K. "A stochastic Model of TCP/IP with Stationary Ergodic Random Losses". *Proc. of ACM-SIGCOMM*, Stockholm, Sweden, pp. 231-242, Aug. 28 - Sept. 1, 2000.

[3] Allman, M., Paxson, V., Stevens, W. "TCP Congestion Control". *RFC 2581*, Network Working Group, IETF, April, 1999.

[4] Arlitt, M. and Williamson, C. "Web server workload characterisation: The search for invariants". *Proc. of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, Philadelphia, May 1996.

[5] Baccelli, F., Hong, D. "TCP is Max-Plus Linear". *Proc. of ACM-SIGCOMM*, Stockholm, Sweden, pp. 219-230, Aug. 28 - Sept. 1, 2000.

[6] Baccelli, F., Liu, Z. "Comparison Properties of Stochastic Decision Free Petri Nets". *IEEE Trans. on Automatic Control*, Vol. 37, pp. 1905-1920, 1992.

[7] Bonald, T. "Comparison of TCP Reno and TCP Vegas: Efficiency and fairness". *Technical report*, RR-3563, INRIA Sophia-Antipolis, 1998, to appear in *Performance Evaluation*.

[8] Crovella, M.E., Bestavros, A. "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes". *IEEE/ACM Transactions on Networking*, 5(6), pp. 835-846, 1997. http://cs-www.bu.edu/faculty/crovella/paper-archive/self-sim/journal-version.ps

[9] Fall, K., Floyd, S. "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP". *Computer Communication Review*, Vol 26, No 3, July, pp. 5-21, 1996.

[10] Floyd, S., Handley, M., Padhye, J. and Widmer, J. "Equation-Based Congestion Control for Unicast Applications". *Proc. of ACM-SIGCOMM*, Stockholm, Sweden, pp. 43-56, Aug. 28 - Sept. 1, 2000.

[11] Gibbens, R. and Kelly, F. "Ressource Pricing and the Evolution of Congestion Control". *Automatica*, 35, pp. 1969-1985, 1999.

[12] Klivansky, S.M., Mukherjee, A., Song, C. "On Long-Range Dependence in NSFNET Traffic". *Technical Report*, College of Computing, Georgia Institute of Technology, 1994. ftp://ftp.cc.gatech.edu/pub/coc/tech_reports/1994/GIT-CC-94-61.ps.Z

[13] Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V. "On the Self-Similar Nature of Ethernet Traffic". *IEEE/ACM Transactions on Networking*, 2(1), pp. 1-15, 1994. ftp://ftp.bellcore.com/pub/wel/tome.ps.Z

[14] Liu, Z., Niclausse, N., Jalpa-Villanueva, C., Barbier, S. "Traffic Model and Performance Evaluation of Web Servers". *Technical Report*, RR-3840, INRIA Sophia-Antipolis, 1999.

[15] Massoulie, L. and Roberts, J. "Bandwidth sharing: objectives and algorithms". *Proc. of INFOCOM*, New York, 1999.

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

[17] Misra, V., Gong, W. and Towsley, D. “Stochastic Differential Equation Modeling and Analysis of TCP Windowsize Behavior”. *Proc. of Performance*, Istanbul, Turkey, October, 1999.

[18] Norros, I. “On the use of Fractionnal Brownian Motion in the theory of connectionless Networks”. *IEEE Journal on Selected Area of Communications*, 13(6), August, 1995.

[19] Padhye, J., Firoiu, V., Towsley, D. and Kurose, J. “Modeling TCP throughput: a simple model and its empirical validation”. *Proc. of ACM-SIGCOMM*, 1998.

[20] Padhye, J., Firoiu, V., Towsley, D. “A Stochastic Model of TCP Reno Congestion Avoidance and Control”. *Technical Report*, 99-02, CMPSCI, Univ. of Massachusetts, Amherst, 1999.

[21] Wright, G. and Stevens, R. “TCP/IP Illustrated”. *Addison Wesley*, Volume 2, 1995.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399