Flow-level Modelling of TCP Traffic Using GPS Queueing Networks

Olivier Brun, Ahmad Al Sheikh and Jean-Marie Garcia
LAAS-CNRS, Université de Toulouse
7, Avenue du Colonel Roche,
Toulouse, France.
Email: {brun, aalsheik, jmg}@laas.fr

***Abstract*—We develop flow-level models of TCP/IP networks. We first consider access network models in a very general framework where Internet subscribers can use several types of TCP-based applications and may have limited peak rates. Using multiclass networks of Generalized Processor Sharing queues (in the sense of Cohen), we derive key performance indicators in closed-form. We then give arguments in favor of a weighted max-min fair (WMM) allocation, with flow weights depending both on the number of TCP connections and on the RTT of the flow, as an approximation of TCP bandwidth-sharing. Finally, we build upon the models developed for access networks to propose an approximate model for the global network, assuming WMM fairness as bandwidth-sharing paradigm. This global network model allows to account for backbone congestion by replacing access links by virtual links whose capacities are computed in order to reflect the probability of congestion on flow path.**

I. Introduction

Internet has become pervasive in the last decade thanks to a succession of drivers, starting with the rise of the World Wide Web, followed by business adoption of Internet applications such as e-mail, and more recently, the use of peer-to-peer file-sharing softwares. The success of the Internet is yet to be amplified in the near future, thanks to three major on-going evolutions. The first one is the progressive transformation of the Internet towards a global communication infrastructure supporting a large variety of multimedia services in addition to traditional document-retrieval applications. The second one concerns the trend towards seamless connectivity between fixed and wireless networks which will allow users to access the same services, no matter what their locations or devices. Finally, the third one is the rapid development of the SaaS (Software as a Service) concept, where a software application is hosted as a service provided to customers across the Internet.

With the Internet getting more and more present in our daily activities, network outages or even significant degradations of the quality of service become less and less tolerable. To avoid network congestions and the resulting service degradations, Internet Service Providers (ISP) need to properly dimension the core network and trunk lines giving the subscribers access to the Internet. A cost-efficient alternative to installing excessive amounts of capacity is to dimension the network on the basis of performance studies allowing to determine the amount of multiplexing gain that can be achieved by exploiting the statistical features of the traffic.

Unfortunately, discrete-event models cannot be used to this end due to their prohibitive computing times. Hybrid models based on differential equations have been developed (see for example [3]), but these models still suffer from the same drawback. In this context, a considerable research effort has been devoted to the development of an Internet teletraffic theory. In contrast with its counterpart for circuit-switched networks, this theory is still in its infancy, although significant progress have been made in recent years. The main difficulty here comes from the elastic nature of most Internet traffics whose throughputs are modulated by the TCP protocol depending on traffic conditions. Two main categories of performance models have been developed: flow level models and packet-level models.

Flow-level models are idealized models that include random flow-level dynamics but use highly simplified models of the bandwidth sharing operated by TCP (see [34] for a survey). Based on the observation that TCP shares bandwidth in an approximately fair way, [18] and [27] proposed the use of the so-called processor-sharing queue as a performance model of access networks. This research direction was pursued to prove the insensitivity of the results to detailed traffic characteristics [5], [7], [8], to address stability issues [12], to account more precisely for the way TCP operates [1], [4], [20] and to extend the modelling framework to any network topology [28], [10], [8], [24].

Packet-level models capture more details of the system (Round Trip Times, buffer size, etc.), but assume a constant number of persistent flows. Here, an important result has been the "square-root formula" for TCP throughput [29], [33], which has allowed the development of several fixed-point algorithms to compute both flow throughputs and link loss probabilities, see e.g. [2] and [13]. An important research effort has also been devoted to the analysis of TCP-like congestion control based on the optimization of some aggregated utility function [19], [21], [25], [30], [36].

Some attemps have also been done to fill the gap between flow-level and packet-level models [22], [23] and to integrate streaming traffics in the modelling framework [32], [11].

The present article is devoted to the flow-level modelling of elastic traffics. In section II, we first revisit access network models in a very general framework where Internet subscribers can use several types of TCP-based applications and have limited peak rates. Using multiclass networks of Generalized Processor Sharing (GPS) queues [15], we derive the stationary joint distribution of the numbers of sessions in each phase as well as its marginals. Key performance indicators are also derived in closed-forms. In section III, we give arguments in favor of a weighted max-min fair allocation, with flow weights depending both on the number of TCP connections and on the RTT of the flow, as an approximation for the bandwidth sharing performed by TCP. In section IV, we then build upon the models developed for access networks to propose an approximate model for the global network. This global network model allows to account for backbone congestion by replacing access links by virtual links whose capacities are computed in order to reflect the probability of congestion on the path of each flow.

II. Access Network Models

We consider here a large number of ADSL or UMTS subscribers distributed over a single geographic area and connected to the backbone network through an access line of capacity C Mbps (see figure 1). The access line is assumed to be the bottleneck link for all the traffic flows generated by users. This link is a full-duplex link, which means that the transmission capacity is C Mbps in both directions.

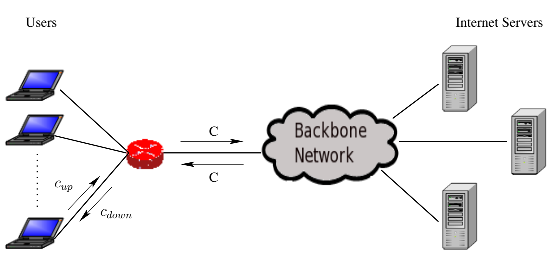

Fig. 1. M Internet users sharing a C bps access link.

The throughput of a user terminal is limited to c_{up} and c_{down} in the upstream and downstream directions, respectively. We let N_{up} (resp. N_{down}) denote the number of TCP connections that can transmit on the uplink (resp. downlink) at their full terminal rate before congestion of this link. For ease of presentation, we assume that $C = N_{up}\, c_{up} = N_{down}\, c_{down}$.

Each individual user can be either idle (inter-session phase) or involved in an on-going session of a TCP-based application (see figure 2). We shall distinguish two broad categories of Internet applications:

- *Bulk data transfer applications:* such applications either upload or download a single file of random size. For instance, uploading a file onto a FTP server or sending an e-mail to a mail server are two examples of bulk data transfer applications generating upload traffic.
- *Interactive applications:* the users of such applications alternate between file downloadings and OFF periods, when the user reads the content of the downloaded document. Typical examples of such applications are Web browsing applications. For such applications, it is assumed that each interactive session begins with a communication phase and ends after an OFF period.

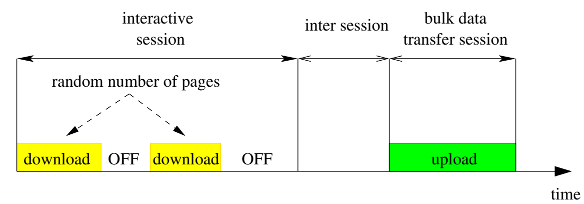

Fig. 2. Activity of an individual user accross time.

We are of course aware that all TCP-based applications generate bidirectionnal traffic, primarily due to the flow of TCP acknowledgment packets, but also due to the exchange of request and replies in the client/server model. However, as an approximation, it can be considered that most Internet applications (although not all) exhibit a main stream of traffic, depending on whether the user wants to send or receive data.

In the sequel, we let $\mathcal{B} = \mathcal{U} \cup \mathcal{D}$ denote the set of bulk data transfer applications, where $\mathcal{U}$ (resp. $\mathcal{D}$) is the subset of these applications generating only upload (resp. download) traffic. Let also $\mathcal{I}$ be the set of interactive Internet applications. Finally, let $\mathcal{R} = \mathcal{B} \cup \mathcal{I}$ be the set of all Internet applications available to the users.

New sessions are initiated by users according to a Poisson process at rate λ. This assumption stems naturally from the fact that individual sessions are independently generated by a large population of users and has been confirmed by measurements [17]. A new session uses application r with probability γ_r. It then transfers one or more files whose sizes are i.i.d. random variables, with mean $1/\alpha_r$ (in Mbits). The numbers of file transfers during a session are also i.i.d. random variables, with mean $1/p_r$. By convention, $p_r = 1$ if $r \in \mathcal{B}$. It is assumed that the number of files downloaded by an interactive application $r \in \mathcal{I}$ is distributed according to a geometric distribution, and we let $1/\beta_r$ denote the average duration of the OFF period for this application.

An important contribution of this paper is to observe that the system can be modeled as an equivalent multiclass network of queues as depicted in figure 3 (class routing probabilities are shown next to the links). This has already been observed previously (see [5] for instance) in a less

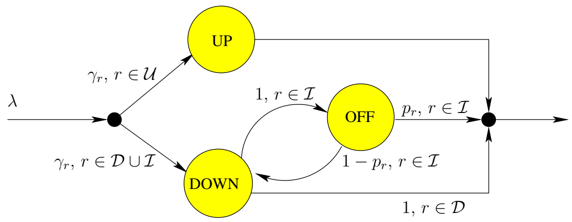

Fig. 3. Equivalent open network of queues.

general context, but in our opinion the idea has not been fully exploited. Customers in this queueing network represent on-going Internet sessions. Exogeneous arrivals represent new Internet sessions started by users, while departures from the network represent the end of Internet sessions. There are multiple classes of customers, each one being associated to an application $r \in \mathcal{R}$. Each node in this queueing network is associated to a possible state of an on-going Internet session: customers at the UP and DOWN nodes represent Internet sessions uploading or downloading a file, while customers at the OFF node represent Internet sessions of interactive applications which are in the OFF period. Let $\mathcal{S} = \{up, down, off\}$ be the set of nodes of this equivalent queueing network. If a customer is at node $s \in \mathcal{S}$, then the corresponding session is said to be in phase s. A session is said to be active if it is in phase $s \in \mathcal{S}_a = \{up, down\}$.

To account for the rate limitations of user terminals, we assume that the UP and DOWN nodes are *Generalized Processor Sharing* (GPS) queues [15]. When there are k connections in phase $s \in \mathcal{S}_a$, an individual connection gets the service rate $f_s(k) = min(c_s, C/k)$, i.e. a connection in phase s is served at its full terminal rate if $k \le N_s$, and according to the ordinary processor sharing discipline otherwise. The solution of the GPS model, as was given by Cohen [15], involves the functions $\Phi_s(k) = \prod_{j=1}^{k} 1/f_s(j)$, $k \ge 1$ with $\Phi_s(0) = 1$. In our case, it yields for $s \in \mathcal{S}_a$,

$$\Phi_s(k) = \begin{cases} (1/c_s)^k & k \le N_s \\ \dfrac{(C/c_s)^{N_s}}{N_s!}\, k!\, (1/C)^k & \text{otherwise} \end{cases}$$

As the OFF node represents a delay, it could be modeled as an infinite server queue. However, it will be convenient in the following to exploit the versatility of the GPS discipline in order to view it as an infinite capacity GPS queue with rate function $f_{off}(k) = 1$, $k \ge 1$ (see [15], section 5). Note that it yields $\Phi_{off}(k) = 1$, $k \ge 1$.

Finally, let $\Omega(s)$ denote the set of applications that can be in phase s: $\Omega(up) = \mathcal{U}$, $\Omega(down) = \mathcal{D} \cup \mathcal{I}$ and $\Omega(off) = \mathcal{I}$. The offered load of each queue s is then easily obtained as $\rho_s = \sum_{r \in \Omega(s)} \rho_s^r$, where $\rho_s^r = \lambda \gamma_r / (p_r \alpha_r)$ for $s \in \mathcal{S}_a$ and $\rho_{off}^r = \lambda \gamma_r / (p_r \beta_r)$. Our main results are stated below. Due to the lack of space, proofs are omitted [1].

1) Joint distribution and its marginals: Let x_s^r denote the number of sessions of application $r \in \Omega(s)$ in phase $s \in \mathcal{S}$. Let us define the following vectors: $\mathbf{x}_s = (x_s^r)_{r \in \Omega(s)}$, $s \in \mathcal{S}$ and $\mathbf{x} = (\mathbf{x}_{up}, \mathbf{x}_{down}, \mathbf{x}_{off})$. Let $\pi(\mathbf{x})$ be the steady-state probability that the system is in state $\mathbf{x}$, if it exists. Finally, let X_s (resp. X_s^r) be the total number of sessions (resp. of application r) in phase s. The queueing network described above can be analyzed as a network of GPS queues. Using theorem 7.2 of [15], we have the following proposition.

Proposition 1: The stationary joint distribution $\pi(\mathbf{x})$ exists provided that $\rho_s < C$, $s \in \mathcal{S}_a$, and it has the following simple product-form,

$$\pi(\mathbf{x}) = \prod_{s \in \mathcal{S}} \pi_s(\mathbf{x}_s)$$

where the probability that node s be in state $\mathbf{x}_s$ is given by,

$$\pi_s(\mathbf{x}_s) = \pi_s(0)\, \Phi_s(|\mathbf{x}_s|) \prod_{r \in \Omega(s)} \frac{(\rho_s^r)^{x_s^r}}{x_s^r!}$$

with the notation $|\mathbf{x}_s| = \sum_r x_s^r$. The probability that there is no connection in phase $s \in \mathcal{S}$ is just $\pi_{off}(0) = e^{-\rho_{off}}$ for $s = off$, while for $s \in \mathcal{S}_a$,

$$\pi_s(0) = \left(\sum_{i=0}^{N_s - 1} \frac{(\rho_s/c_s)^i}{i!} + \frac{(\rho_s/c_s)^{N_s}}{N_s!} \frac{C}{C - \rho_s} \right)^{-1}$$

Moreover, for $s \in \mathcal{S}$ and $r \in \Omega(s)$,

$$Pr[X_s = k] = \pi_s(0)\, \Phi_s(k)\, \frac{(\rho_s)^k}{k!}$$

$$Pr[X_s^r = k] = \pi_s(0)\, \frac{(\rho_s^r)^k}{k!} \sum_{i=k}^{\infty} \Phi_s(i)\, \frac{(\rho_s - \rho_s^r)^{i-k}}{(i-k)!}$$

Note that the above results, as well as all the results presented in this section, are insensitive to detailed traffic characteristics. Observe that the probability distribution of X_s is the solution of the $M/M/C/\infty$ queue for $s \in \mathcal{S}_a$. Therefore, the probability B_s of congestion of the access line in direction s is given by the well-known Erlang C formula,

$$B_s = Pr\left[X_s \ge N_s\right] = \pi_s(0)\, \frac{(\rho_s/c_s)^{N_s}}{N_s!} \frac{C}{C - \rho_s}$$

2) Key Performance Measures: Let D_s (resp. D_s^r) be the duration of a file transfer (resp. of application r) in direction $s = up, down$. We have the following results regarding the expected numbers of sessions and sojourn times in each phase.

[1] Note to reviewers: an extended version of this paper including proofs is available at www.laas.fr/∼brun/.

Proposition 2: The mean number of sessions in phase $s \in \mathcal{S}_a$ is given by the expected number of customers in the corresponding $M/M/C/\infty$ queue,

$$E\left[X_s\right] = \rho_s \left(\frac{1}{c_s} + \frac{B_s}{C - \rho_s}\right)$$

whereas $E[X_{off}] = \rho_{off}$. Moreover, the mean number of sessions of application r in phase s is $E[X_s^r] = (\rho_s^r/\rho_s)\, E[X_s]$ for $s \in \mathcal{S}$ and $r \in \Omega(s)$.

According to Little's law, $E[D_s] = E[X_s]/\lambda$ and $E\left[D_s^r\right] = \lambda E\left[D_s\right]/(\rho_s \alpha_r)$. Moreover, from equation (7.27) of [15], the mean transfer time of a x Mbits file in direction $s \in \mathcal{S}_a$ is $E\left[D_s \,|\, size = x\right] = \lambda x E\left[D_s\right]/\rho_s$.

Let the random variable T_s denote the data traffic in Mbps on the access line in direction s. Let also $R_s = T_s/X_s$ be the instantaneous throughput one connection gets in direction s when $X_s > 0$.

Proposition 3: The average traffic in direction $s \in \mathcal{S}_a$ is,

$$E\left[T_s\right] = \sum_{k=1}^{\infty} k\, min(c_s, C/k)\, Pr[X_s = k] = \rho_s$$

which is independant of the capacity, as expected. The square-root coefficient of variation of T_s is just $(c_s/\rho_s)\,(1 - B_s)$. The instantaneous throughput per session in direction $s = up, down$ is c_s with probability $(1 - \rho_s B_s/C)/(1 - \pi_s(0))$ and C/k with probability $Pr[X_s = k]/(1 - \pi_s(0))$ for $k > N_s$. Its average value is given by,

$$E\left[R_s\right] = c_s \frac{1 - \rho_s B_s/C}{1 - \pi_s(0)} + (C - \rho_s)\, Z$$

$$\text{where, } Z = \frac{B_s (C/\rho_s)^{N_s}}{1 - \pi_s(0)} \left[ln\left(\frac{C}{C - \rho_s}\right) - \sum_{k=1}^{N_s} \frac{(\rho_s/C)^k}{k} \right]$$

III. Bandwidth-sharing in IP networks

If there were no congestion in the backbone network, the appropriate models would be those introduced in section II. In section IV, we show how to extend these models in order to account for congestion in the backbone network. To this end, we analyse here the bandwidth-sharing performed by TCP in a deterministic setting and propose a simple model of it. Our bandwidth-sharing model is related to the notion of max-min fairness. This notion was introduced as a design objective for communication networks in [6]. The principle of max-min fairness is to allocate bandwidth resources in such a way that the data rate of a flow cannot be increased without decreasing the data rate of a flow with a lower data rate. In this section, we give arguments in favor of a weighted max-min fair (WMM) allocation, with flow weights depending both on the number of TCP connections and on the RTT of the flow, as an approximation of TCP bandwidth sharing.

A. Notations and assumptions

Let us consider a data network with a set $\mathcal{L}$ of links and a set $\mathcal{F}$ of TCP flows, numbered $1, \ldots, K$. Let C_l be the capacity of link l, and Γ_l be the set of flows passing through this link.

Let n_f, θ_f and π_f be the number of TCP connections, the round-trip time (RTT) and the end-to-end path of flow f, respectively. Connections are assumed to be persistent TCP connections, so that the number of connections of each flow is constant. Moreover, it is assumed that the contribution of queueing delays to the RTTs is negligible (large delay-bandwidth product network [2]). Finally, let $\phi_f(\mathbf{n})$ be the aggregated throughput of flow f, where $\mathbf{n} = (n_1, \ldots, n_K)$.

In the sequel, we shall make the following assumptions:

- *Assumption 1*: the packet-loss probability is the same for all the TCP flows "bottlenecked" on the same link.
- *Assumption 2*: if link l is not the bottleneck link for flow f, then the packet-loss probability of flow f on link l is negligible.

Assumption 1 is justified if RED is used as queue management policy [16]. Assumption 2 is clearly more debatable. It merely states that if link l is not the bottleneck for flow f, then link l has no influence on its rate allocation, even if it is fully utilized. In other words, even if the capacity of link l was increased, the amount of bandwidth allocated to flow f would not increase, since its data rate is limited elsewhere.

B. Single bottleneck case

Let us first consider the case where a link $l \in \mathcal{L}$ is the single bottleneck link for all TCP flows f crossing it. In this case, according to assumption 1, the end-to-end loss rate is equal to p for all flow f, where p is the loss rate of the bottleneck link l. Moreover, it is known that the bottleneck link will be saturated. Using the "square-root formula" [29], [33] for the throughput of each individual connection of flow i,

$$\sum_{f \in \Gamma_l} \phi_f(\mathbf{n}) = \sum_{f \in \Gamma_l} \frac{n_f}{\theta_f} \frac{\sqrt{cst}}{\sqrt{p}} = C_l.$$

Solving for p, we get:

$$p = \frac{1}{C_l^2} \left(\sum_{k \in \Gamma_l} \frac{n_k \sqrt{cst}}{\theta_k} \right)^2$$

It is then easy to show that,

$$\phi_f(\mathbf{n}) = \frac{\nu_f}{\sum_{k \in \Gamma_l} \nu_k} C_l \quad , \; f \in \Gamma_l,$$

where $\nu_f = n_f/\theta_f$ is the ratio of the number of TCP connections over the RTT for flow f. Therefore, when a link is the single bottleneck link for all TCP flows crossing it, its capacity is shared among TCP flows according to a WMM allocation with the weights ν_f, $f \in \mathcal{F}$.

If we now assume that link l is the single bottleneck link for only a subset $B_l \subset \Gamma_l$ of the TCP flows crossing it (i.e the loss probability is negligible at this node for all the other flows according to assumption 2), we have,

$$\phi_f(\mathbf{n}) = \frac{\nu_f}{\sum_{k \in B_l} \nu_k} \left[C_l - \sum_{k \in \Gamma_l - B_l} \phi_k(\mathbf{n}) \right] \quad , \; f \in B_l,$$

The above equation shows that the residual capacity of the link (the capacity left by the flows for which the link is not a bottleneck) is shared among the TCP flows for which the link is the bottleneck according to a WMM allocation.

C. Generalization

The above results suggest that, when the number of bottleneck links will be sufficiently small, a WMM allocation can be a good approximation of what TCP does.

Weighted max-min fairness can be obtained by using an algorithm of progressive filling (the so-called water-filling algorithm [31]). The algorithm starts with all flow rates equal to 0 and grows all rates together at pace ν_f for flow f, until one or several link capacity limits are hit. The rates for the flows that use these links are not increased any more, and the algorithm continues increasing the rates for other flows. Finally, when the algorithm terminates, all flows have been stopped at some time and thus have a bottleneck link. The resulting allocation is a WMM allocation. Note that this algorithm can easily be extended to account for the rate limitations of the terminals.

D. Benchmark examples

To assess the accuracy of the WMM approximation, we compare it with the distributed rate allocation of TCP. The latter is determined using the ns-2 simulator. Comparisons also include the results obtained with the “network calculus” fixed-point algorithm of [2] and with the utility function approach of [26]. In all our experiments, we have used the New Reno TCP version and have chosen RED as a queue management policy. The packet size is 1024 Bytes, the propagation delay of links is 10 ms and the buffer sizes are fixed to 1024 packets. Peak rates of connections are unlimited.

Let us consider the networks shown in figures 4 and 5. Results are reported in table I for the network of figure 4 in the cases of 20 and 30 connections per flow. Results are reported in table II for the other network in the case of 10 connections per flow.

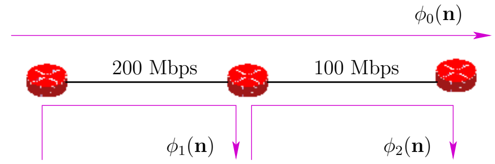

Fig. 4. Two link network with cross traffics.

TABLE I
RATE ALLOCATIONS FOR THE TOPOLOGY OF FIGURE 4.

	20 cnx/flow			30 cnx/flow		
	ϕ_0	ϕ_1	ϕ_2	ϕ_0	ϕ_1	ϕ_2
Weighted Max-Min	33.3	166.6	66.6	33.3	166.6	66.6
Network Calculus	32.9	168.4	71.3	34.4	168.7	74.8
TCP Utility	31.6	168.3	68.3	31.6	168.3	68.3
ns2	31.4	167.6	68.3	31.3	167.6	68.4

For both examples, the WMM allocation can be considered as a fairly good approximation of the TCP rate allocation. We of course do not claim that this WMM allocation should be viewed as an accurate presentation of TCP behavior, but observe that it often yields an accurate degree of approximation. It can be expected that the quality of the approximation is even better in a dynamic setting, in particular if connections have limited peak rates, since in this case the probability of several links being fully utilized is typically low. Moreover, weighted max-min fairness has a couple of attractive features. First, it nicely captures the impact of round-trip times on throughputs. Second, the rate allocation can be efficiently computed using the water-filling algorithm, an obviously desirable feature for iterative performance evaluation methods.

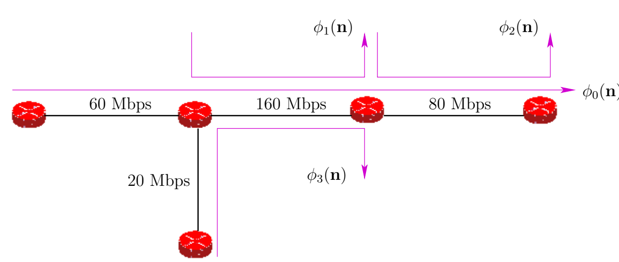

Fig. 5. Four link network with cross traffics.

TABLE II
RATE ALLOCATIONS FOR THE TOPOLOGY OF FIGURE 5.

	ϕ_0	ϕ_1	ϕ_2	ϕ_3
Weighted Max-Min	20	120	60	20
Network Calculus	18.5	121.9	62.5	20.5
TCP Utility	18.3	121.6	61.6	20
ns2	18.5	120.2	61	19.5

IV. Global network model

In this section, we consider the situation depicted in Figure 6, where users located in K different access sites are connected to a backbone network by access links of capacity $C_1, \ldots, C_K$ and generate traffic from a set of Internet servers. For ease of presentation, it is assumed below that the population of each access site is infinite and that all traffics go from the server farm to the access sites. However, the modelling approach can easily be extended to the case of multiple sources and multiple destinations with finite or infinite user population for each access site, and with upload and download traffics.

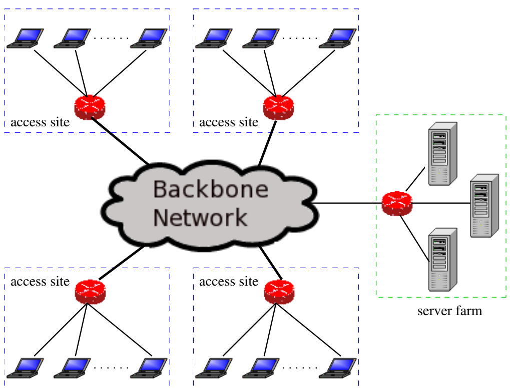

Fig. 6. Internet users downloading data from Internet servers.

If there were no congestion in the backbone network, the appropriate models would be those introduced in section II. We shall show how to extend these models in order to account for the congestion in the backbone network.

A. Fixed-point algorithm

Due to congestion in the backbone network that occurs from time to time, each access site i cannot send data at the full rate C_i, but at a lower rate. Let C_i^* be the average transmission capacity seen by connections of flow i. The main idea is then to replace the access lines and the backbone network by virtual access links of capacity C_i^* connecting directly the access sites to the server farm.

Let the random variable X_i denote the number of active sessions in access site i and let $\mathbf{X} = (X_1, \ldots, X_K)$. According to proposition 1, the probability of $X_i = n$ is,

$$\pi_n^i(C_i^*) = \pi_0^i(C_i^*)\,\Phi_i(n, C_i^*)\,\frac{(\rho_i)^n}{n!} \quad , \; n = 0, 1, \ldots$$

Let $\overline{T}_i(\mathbf{C}^*)$ be the average traffic generated by site i, where $\mathbf{C}^* = (C_1^*, \ldots, C_K^*)$. Using weighted max-min fairness as an approximation of TCP bandwidth-sharing, we have,

$$\overline{T}_i(\mathbf{C}^*) = E\left[\phi_i\left(\mathbf{X}\right)\right] = \sum_{\mathbf{n}\in\mathbb{N}^K} \phi_i(\mathbf{n})\, Pr\left[\mathbf{X} = \mathbf{n}\right]$$

If there is enough capacity in the backbone network, the probability that the rate of an access site be limited by a backbone link will be low. In this case, the random variables $X_1, \ldots, X_K$ can be considered as independant, and we can do the following approximation,

$$\overline{T}_i(\mathbf{C}^*) = \sum_{\mathbf{n}\in\mathbb{N}^K} \phi_i(\mathbf{n}) \prod_{j=1}^{K} \pi_{n_j}^j(C_j^*) \tag{1}$$

However, if we interpret C_i^* as the capacity of an equivalent access line, then $\overline{T}_i(\mathbf{C}^*)$ can also be written as follows:

$$\overline{T}_i(\mathbf{C}^*) = \sum_{k=1}^{\infty} k\, min(c_i, C_i^*/k)\, \pi_k^i(C_i^*) \tag{2}$$

which is just ρ_i in the infinite population case. The capacity C_i^* should be such that expressions 1 and 2 coincide. A solution vector $\mathbf{C}$ can then be find using algorithm 1.

algorithm 1 Fixed-Point Algorithm

procedure Approximation
  $t = 0$ and $C_i^0 = C_i$, $i = 1, \ldots, K$
  while not convergence **do**
    Compute $\pi_n^i(C_i^t)$, $n \geq 0$, $i = 1, \ldots, K$,
    Evaluate $\overline{T}_i(\mathbf{C}^t) = \sum_{\mathbf{n}} \phi_i(\mathbf{n}) \prod_{j=1}^{K} \pi_{n_j}^j(C_j^t)$
    Find C_i^{t+1} as the unique solution in x of,

$$\overline{T}_i(\mathbf{C}^t) = \sum_{k=1}^{\infty} k\, min(c_i, x/k)\, \pi_k^i(C_i^t)$$

    $t = t + 1$
  end while
end procedure

B. Results

As a benchmark example, let us consider the network of figure 7 where C_1=20 Mbps and C_2=10 Mbps. The capacity C of the common backbone link is taken for several values, ranging from 17.5 Mbps to 30 Mbps. All link delays are set to 1 ms. Users are located in access sites 1 and 2 and download data from the web server using an interactive aplication, but with different behaviors for sites 1 and 2 (see table III). All probability distributions are negative exponential distributions. Peak rates are set to $c_1 = 3$ Mbps for users of site 1 and to $c_2 = 6$ Mbps for users of site 2.

We compare below the results obtained with our proposed approximation with those produced by a home-made event-driven simulator implementing the WMM allocation as

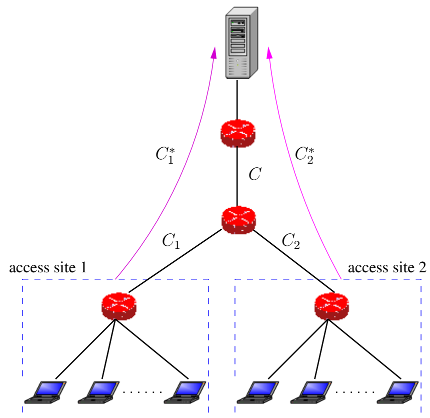

Fig. 7. 3-link network.

TABLE III
AVERAGE VALUES FOR ACCESS SITES 1 AND 2.

Parameter	site 1	site 2
File size (MBytes)	0.5	1
Arrival rate (sessions/s)	1.25	0.5
Number of pages (pages/session)	2	1.5
Off time (seconds)	30	45

TABLE IV
RESULTS FOR $E[X_1]$ AND $E[X_2]$

C	Simulation		Approximation	
	$E[X_1]$	$E[X_2]$	$E[X_1]$	$E[X_2]$
17.5	7.882 ± 0.5	4.665 ± 0.33	9.559	7.034
20.0	4.430 ± 0.09	2.517 ± 0.09	4.110	2.676
22.5	3.713 ± 0.05	2.065 ± 0.06	3.642	2.182
25	3.514 ± 0.05	1.907 ± 0.057	3.492	1.933
27.5	3.454 ± 0.03	1.804 ± 0.044	3.440	1.846
30	3.416 ± 0.027	1.770 ± 0.057	3.422	1.797

bandwidth-sharing paradigm. Table IV compares the expected number of active sessions for flows 1 and 2 (99% confidence intervals are indicated for simulations). Table V presents the same comparison for the mean instantaneous throughput of an individual connection. The approximation yields acceptable results for $C \geq 20$ Mbps. Note that for $C = 20$ Mbps, the independence assumption is far from being satisfied since the proportion of time the common link limits the rate of flow 1 is 38%, and it is 47% for flow 2. The quality of the approximation is similar when the delay of the access link of site 1 is increased to 2 ms (i.e. when $\theta_1 = 6$ms and $\theta_2 = 4$ms).

The algorithm is not limited to average values, but yields approximations of probability distributions as well. Let us consider the case $C = 25$ Mbps. In this case, the proportion of time the common link limits the rate of flow 1 is 8.6%, and it is 15.8% for flow 2. Figures 8 and 9 plot the probability distributions of the number of active sessions of site 1 and of the total number of sessions of site 2, respectively. Figure 10 plots the probability distribution of the instantaneous rate per connection allocated to flow 2.

TABLE V
RESULTS FOR $E[R_1]$ AND $E[R_2]$

C	Simulation		Approximation	
	$E[R_1]$	$E[R_2]$	$E[R_1]$	$E[R_2]$
17.5	1.91	2.17	1.74	1.94
20.0	2.62	3.38	2.73	3.56
22.5	2.86	3.97	2.89	4.01
25	2.94	4.28	2.94	4.31
27.5	2.96	4.45	2.96	4.42
30	2.97	4.52	2.97	4.49

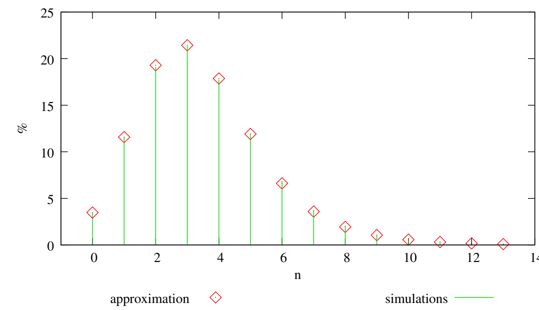

Fig. 8. Probability distribution of the number of active sessions in site 1.

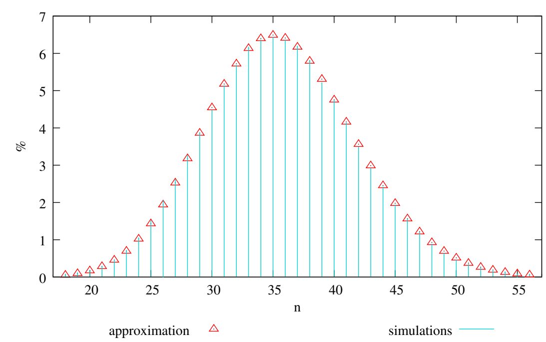

Fig. 9. Probability distribution of the total number of sessions in site 2.

We conduced the same experiments using two-phase hyper-exponential distributions with balanced means for file sizes. Average file sizes are unchanged and the square root coefficient of variations of these distributions are set to 2. For $C \geq 20$, simulation results are within 1.5% of those reported in tables IV and V, suggesting that our model is approximately insensitive to detailed traffic characteristics.

V. CONCLUSION

We have shown how queueing networks of GPS queues can be used to derive key performance indicators in the absence of congestion in the backbone network. To account for the congestion occuring from time to time in the backbone network, we proposed an approximate model, assuming weighted max-min fairness as bandwidth-sharing paradigm. This global model can be used for network dimensioning. Future works

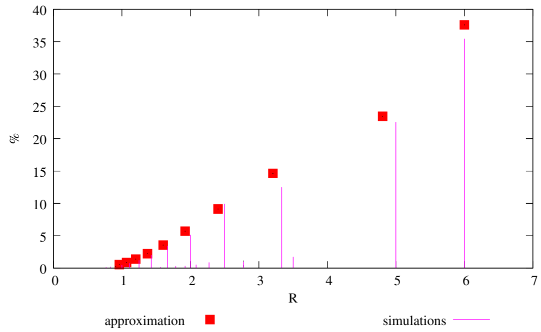

Fig. 10. Instantaneous rate per connection of flow 2.

include the analysis of the weighted max-min fair allocation introduced in this paper in a stochastic context.

ACKNOWLEDGMENT

The authors would like to thank Urtzi Ayesta and Balakrishna Prabhu for helpful discussions.

REFERENCES

[1] D. Abendroth, H. van den Berg and M. Mandjes, *A Multiple Time-Scale Model for TCP Bandwdth Sharing Under User Heterogeneity*, in Proc. of Networking 2005 (R. Boutaba et al. Eds), LNCS 3462, 2005.

[2] E. Altman, K. Avrachenkov and C. Barakat, *TCP Network Calculus: The case of large delay-bandwidth product*, in Proc. of IEEE INFOCOM, 2002.

[3] F. Bacelli and D. Hong, *Flow Level Simulation of Large IP Networks*, in Proc. of IEEE INFOCOM, 2003.

[4] J.V.L. Beckers, I. Hendrawan, R.E. Kooij and R.D. van der Mei, *Generalized Processor Sharing Performance Models for Internet Access Lines*, in Proc. of 9th IFIP Conference, 2001.

[5] S. Ben Fredj, T. Bonald, A. Proutiere, G. Régnié, J.W. Roberts, *Statistical Bandwidth Sharing: A Study of Congestion at Flow Level*, in Proc. of SIGCOMM'01, 2001.

[6] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.

[7] T. Bonald and A. Proutière, *Insensitivity in processor-sharing networks*, Performance Evaluation 49, 2002.

[8] T. Bonald and A. Proutière, *Insensitive Bandwidth Sharing in Data Networks*, Queueing Systems: Theory and Applications 44(1), 2003.

[9] T. Bonald and A. Proutière,J.W. Roberts and J. Virtamo, *Computational aspects of balanced fairness*, in Proc. of ITC 18, 2003.

[10] T. Bonald and A. Proutière, *On performance bounds for balanced fairness*, Performance Evaluation 55, 2004.

[11] T. Bonald and A. Proutière, *On performance bounds for the integration of elastic and adaptive streaming flows*, in Proc. of ACM SIGMETRICS/Performance, 2004.

[12] T. Bonald, L. Massoulié, A. Proutière and J. Virtamo *A queueing analysis of max-min fairness, proportional fairness and balanced fairness*, Queueing Systems 53, 2006.

[13] T. Bu and D. Towsley, *Fixed point approximations for TCP behavior in an AQM network*, in Proc. of SIGMETRICS'01, 2001.

[14] J.P. Buzen, *Computational algorithms for closed queueing networks with exponential servers*, Comm. of the ACM 16 (9), 1973.

[15] J.W. Cohen, *The Multiple Phase Service Network with Generalized Processor Sharing*, Acta Informatica 12. 1979.

[16] S. Floyd and V. Jacobson, *Random Early Detection (RED) gateways for Congestion Avoidance*, IEEE/ACM Trans. Network. 1(4), 1993.

[17] S. Floyd and V. Paxson, *Difficulties in simulating the Internet*, IEEE/ACM Trans. Network. 9 (4), 2001.

[18] D.P. Heyman, T.V. Lakshman and A. Neidhardt, *A New Method for Analysing Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet*, Proc. of SIGMETRICS'97, 1997.

[19] F.P. Kelly, A. Maulloo and D. Tan, *Rate control for communication networks: Shadow prices, proportional fairness and stability*, Journal of the Operat. Res. Society 49, 1998.

[20] A. Kherany and A. Kumar, *Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet*, Proc. of IEEE INFOCOM, 2002.

[21] S. Kunniyur and R. Srikant, *End-to-end congestion control schemes: Utility functions, Random losses, ECN marks*, Proc. of IEEE INFOCOM, 2000.

[22] P. Lassila, H. van den Berg, M. Mandjes and R. Kooij *An integrated packet/flow model for TCP performance analysis*, Proc. of ITC 18, Elsevier, Amsterdam, 2003.

[23] P. Lassila, M. Mandjes, *A Multi-level TCP Model with Heterogeneous RTTs*, Proc. of Networking. 2004 (R. Mitrou et al. Eds), LNCS 3042, 2004.

[24] P. Lassila and J. Virtamo, *Access network dimensioning for elastic traffic based on flow-level QoS*, Networking Laboratory, Helsinki University of Technology, 2005

[25] S.H. Low and D.E. Lapsley, *Optimization flow control I: Basic Algorithm and Convergence*, IEEE/ACM Trans. on Networking 7(6), 1999.

[26] S.H. Low, *A duality model of TCP and queue management algorithms*, IEEE/ACM Trans. on Networking 11(4), 2003.

[27] L. Massoulié and J.W. Roberts *Bandwidth sharing and admission control for elastic traffic*, Telecommunication Systems 15, 2000.

[28] L. Massoulié and J.W. Roberts *Bandwidth sharing: Objectives and algorithms*, IEEE/ACM Trans. Netw. 10(3), 2002.

[29] M. Mathis, J. Semke, J. Mahdavi and T. Ott, *The macroscopic behavior of the TCP congestion avoidance algorithm*, Computer Communication Review 27, 1997.

[30] J. Mo and J.C. Walrand, *Fair end-to-end window-based congestion control*, IEEE/ACM Trans. Netw. 8(5), 2000.

[31] D. Nace, M. Pioro, *A Tutorial on Max-Min Fairness and its Applications to Routing and Load-Balancing in Telecommunication Networks*, to appear in IEEE Jour. Comm. Surveys and Tutorials.

[32] R. Núñez Queija, J.L. van den Berg and M. Mandjes, *Performance evaluation of strategies for integration of elastic and stream traffic*, Proc. of ITC 16, 1999.

[33] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, *Modeling TCP Throughput: a simple Model and its Empirical Validation*, Proc. of ACM SIGCOMM'98, 1998.

[34] J.W. Roberts, *A survey on statistical bandwidth sharing*, Computer Networks 45, 2004.

[35] C.H. Sauer, *Computational algorithms for state-dependent queueing networks*, ACM Trans. Computer Systems 1 (1), 1983.

[36] M. Vojnovic, J.-Y. Le Boudec and C. Boutremans, *Global fairness of additive-increase and multiplicative-decrease with heterogeneous rountrip times*, Proc. of IEEE INFOCOM, 2000.