

# An integrated packet/flow model for TCP performance analysis

Pasi Lassila<sup>ab\*</sup>, Hans van den Berg<sup>ac\*</sup>, Michel Mandjes<sup>ad\*</sup> and Rob Kooij<sup>c\*</sup>

<sup>a</sup>University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands

<sup>b</sup>Helsinki University of Technology, P.O.Box 3000, FIN 02015 HUT, Finland

<sup>c</sup>TNO Telecom, P.O. Box 421, 2260 AK Leidschendam, the Netherlands

<sup>d</sup>CWI, P.O. Box 94079, 1090 GB, Amsterdam, the Netherlands

Processor sharing (PS) models for TCP behavior nicely capture the bandwidth sharing and statistical multiplexing effect of TCP flows on the flow level. However, these ‘rough’ models do not provide insight into the impact of packet-level parameters (such as round trip time and buffer size) on, e.g., throughput and flow transfer times. This paper proposes an integrated packet/flow-level model: it exploits the advantages of PS approach on the flow level and, at the same time, it incorporates the most significant packet-level effects.

## 1. INTRODUCTION

The vast majority of traffic on the Internet relates to the transfer of jobs (web pages, audio/video downloads, file transfers, etc.) coordinated by TCP. TCP has been designed to support efficient and reliable transmission of *elastic* jobs, i.e., jobs that tolerate (some) variations in the throughput. By noticing packet loss, TCP is implicitly provided with information about the level of congestion along the path through the network. Based on this information, the traffic sending rate is adapted. During periods of low utilization, TCP increases its transmission rate, whereas during congestion it decreases its rate. The main TCP performance measures are throughput and transfer delays. TCP’s widespread use and complex behavior explain the search for simple, yet accurate, mathematical models for TCP performance analysis, in terms of the traffic parameters (job size distribution, interarrival times, etc.) and network parameters (link rates, buffer sizes, etc.). Currently, two successful approaches for TCP performance analysis are available, both having their strong and weak aspects. We will refer to these models as (i) flow-level models, and (ii) packet-level models.

*Flow-level models* take into account the dynamics related to the arrival and departure of TCP flows. It is assumed that the available link bandwidth is divided equally among the jobs. Loosely speaking, this means that there is instantaneous feedback from the

---

\*The work of P. Lassila is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs, the Netherlands. Work of other authors partially funded by the Dutch Ministry of Economic Affairs under the programme ‘Technologische Samenwerking ICT Doorbraakprojecten’, project TSIT 2031 EQUANET.

network to the sources, in that the transmission rates adapt instantly to changes in the number of flows present. The above modeling enables an analysis through *processor sharing queues*, discussed in great generality by Cohen in, e.g., [3]. These models were recently ‘rediscovered’ in the context of TCP, see for instance [2], [7], [9], [11], and [12]. They nicely capture the flow-level dynamics, but do not give insight into the impact of packet-level parameters, e.g., buffer sizes and round trip times (RTTs), on throughput and flow transfer delays. A notable feature in these models is that throughput and average job transfer delay depend on the job size distribution *only through the mean job size*.

*Packet-level models* capture more details of the system (RTTs, buffer size, etc.), but do not take into account flow-level dynamics (i.e., the dynamics related to the arrival and departure of flows). In other words, these models assume a fixed number of persistent flows. Early results in this area were by Mathis *et al.* [8] and Padhye *et al.* [10]. Notably, in these papers a straightforward relation is derived between throughput on the one hand, and packet loss probability and RTT on the other hand. Because usually the packet loss probability is not known *a priori* (but is rather determined ‘endogenously’ by the feedback system), some additional effort is required to compute the throughput. This can be done by expressing the loss probability as a function of the input rate (which is closely related to the throughput), thus obtaining a fixed-point iteration scheme; see also [1], [4], and [6].

*Our contribution* is the development of an integrated packet/flow-level model for the analysis of a bottleneck link in a TCP/IP network environment, combining the attractive features of both flow-level and packet-level models. Our procedure roughly works as follows. First we compute, for a given number of flows, say  $n$ , in the system, the throughput  $t_n$ . This is done by using a packet-level model, hence RTTs and buffer size are reflected by the  $t_n$ . These  $t_n$  are the inputs for the flow-level model. We extensively validate our approach by using ns2. More specifically, we experiment with different system parameters and discuss under which conditions our approximation leads to accurate results.

Related work is done by Gibbens *et al.* [4]. For a fixed number of sources, they can compute the throughput, just as we do (although in [4] this is even done in a network setting rather than just a single node). To take into account the fluctuating number of concurrent flows  $N$ , they assume  $N$  to be either Poisson or geometrically distributed. In other words, in [4]  $N$  is chosen (exogenously), whereas we derive the distribution of  $N$  (endogenously) from our model, by applying the flow level on top of the packet level. The approach followed by Massoulié and Roberts [7] is related to ours, but the feedback modeling on the packet level is less detailed, as no actual queueing model is used to obtain the packet loss probabilities.

The paper is organized as follows. Section 2 describes in more detail the two approaches introduced above (i.e., the packet-level and flow-level models). In Section 3, we present the integrated packet/flow-level model. The model is validated in Section 4, where we compare the accuracy of the model with results obtained by using the ns2 network simulator. Conclusions and suggestions for future research are given in Section 5.

## 2. MODELING APPROACH AND PRELIMINARIES

The model of this paper considers a network link with speed  $C$  and buffer space of  $K$  packets. Flows arrive at the link according to a Poisson process with rate  $\lambda$ , whereas the

length of the flows (in packets) are i.i.d. samples from a (general) common distribution with mean  $1/\mu < \infty$ . To ensure stability, the load of the system must satisfy  $\lambda/\mu < C$ . We assume that the users' transmission rates are restricted to  $r$  (due to, e.g., a limited access rate or modem speed). We also assume that all flows have the same round trip time, RTT. Packet transmissions of the flows are coordinated by TCP's window-based congestion control mechanism. Note that the maximum window size  $W_{max}$  and the RTT also put a limit on the users' transmission rate, i.e.,  $r \leq W_{max}/RTT$ . In the following subsections we subsequently describe the two 'sub-models' that form the components of our integrated packet/flow model.

## 2.1. Flow-level approach

Here a general overview is presented of known results in the context of processor sharing (PS) queueing systems. Specifically, we describe a generic variant of PS, commonly referred to as *generalized processor sharing*, or GPS.

- *The GPS model, ordinary PS.* We study a Poissonian stream of jobs, feeding into a server of capacity  $C$ . The job arrival rate is  $\lambda$ , whereas job sizes are i.i.d. with mean  $1/\mu < \infty$ . When  $n$  jobs are in the system, they are allowed to transmit their data at a (joint) rate  $r_n$ ; the server divides this capacity equally among the jobs, so each of them is assigned a share  $r_n/n$ .

We present the solution of this GPS model, as was given by Cohen [3]. First define

$$\phi(n) := \begin{cases} \lambda \cdot (\mu r_n)^{-1}, & \text{for } n \in \mathbb{N}, \\ 1, & \text{for } n = 0, \end{cases} \quad \text{and} \quad \psi(n) := \prod_{i=0}^n \phi(i).$$

Then the distribution of the number of jobs in the system,  $N$ , is given by

$$\mathbb{P}(N = n) = \frac{\psi(n)}{\sum_{m=0}^{\infty} \psi(m)}. \tag{1}$$

Now that we are able to compute the mean number of jobs in the system  $\mathbb{E}N$ , the mean transmission delay  $\mathbb{E}D$  follows from Little's law:  $\mathbb{E}N = \lambda \cdot \mathbb{E}D$ . In fact, the mean delay for a job of given size  $x$  can be computed explicitly – notably, this quantity grows *linearly* in  $x$ . The most important feature of the GPS model, however, is that these performance measures are *insensitive* with respect to the job size distribution. More precisely, they only depend on this distribution through the mean  $1/\mu$  (see [3] for details).

The special case in which  $r_n := C$  (i.e., in principle all bandwidth is available to the users) is usually referred to as the *ordinary PS model*, in which case  $N$  is geometrically distributed with  $\rho := \lambda(\mu C)^{-1}$ :  $\mathbb{P}(N = n) = \rho^n(1 - \rho)$ .

- *TCP and the GPS model; ideal GPS.* If the focus is to apply the GPS model to TCP, a few remarks need to be made. In principle, the task of TCP is to fairly share the available capacity, and the GPS model discussed above indeed does so. The ordinary processor sharing case, however, assumes that a single TCP flow can utilize the whole capacity, which is not very realistic.

The GPS model, however, also nicely covers a situation in which the users are restricted by (common) peak rates  $r$ , by choosing  $r_n := \min\{C, nr\}$ . In this case the 'left tail' of the distribution of  $N$  has a 'Poisson-shape', whereas the 'right-tail' resembles a geometric

distribution. The peak rate could relate to, say, the physical rate of the access link, or the receiver’s advertised window size.

It is noticed that the GPS model with  $r_n := \min\{C, nr\}$  assumes, in fact, an infinite packet buffer and ‘ideal’ adaptation of the flow rates to their fair share of the link rate (instantaneous rate adaptation without packet losses). Thus, in particular, the model does not take into account the impact of the buffer size on the TCP flow throughput, and the effect of round trip times. Application of GPS models will in many cases (e.g. when buffer sizes are relatively small) lead to TCP performance estimates that are too optimistic, see, e.g., [12]. In the sequel, we will refer to the GPS model with  $r_n = \min\{nr, C\}$  as the *ideal GPS model* (to distinguish this from the integrated model that will be introduced in Section 3).

## 2.2. Packet-level approach

Now consider the situation in which a constant number, say  $n$ , of persistent TCP flows share a link. Assume for ease that the users have identical round trip times RTT and peak rates  $r$ . The problem of determining the mean throughput  $t_n$  (expressed in packets/sec) as a function of the loss rate  $p$  and RTT has received considerable attention in the literature [5], [8], and [10]. By now, it is well known that the (aggregate) throughput  $t_n$  approximately satisfies

$$t_n = \min \left\{ nr, \frac{n\Gamma}{\text{RTT}} \frac{1}{\sqrt{p}} \right\}, \quad (2)$$

where the value of the constant term  $\Gamma$  depends on the specific modeling assumptions.

To utilize the above result, one needs to somehow determine the loss rate  $p_n$  (note that the subscript  $n$  is added to express the dependence on  $n$ ). A simple solution is the following iterative method, see, e.g., [1], and [6]. The above relation (2) describes the throughput  $t_n$  as a function of  $p_n$ , but, in return,  $p_n$  is also affected by the amount of traffic offered to the link, i.e.,  $t_n$ . Assuming that packet arrivals are Poissonian, then one could argue that  $p_n$  should equal the loss probability in an M/D/1 queue equipped with buffer  $K$  and link rate  $C$ , and Poisson arrival rate  $t_n$ ; hence  $p_n$  is increasing in  $t_n$ . Thus,

$$t_n = \min \left\{ nr, \frac{n\Gamma}{\text{RTT}} \frac{1}{\sqrt{p(t_n)}} \right\}, \quad (3)$$

where  $p(t_n) = p_n$  is the loss probability in an M/D/1/ $K$  queue with packet arrival rate  $t_n$ . The existence and uniqueness of the fixed point of this equation is ensured by the fact that the right hand side decreases in  $t_n$  and is continuous in  $t_n$ .

## 3. INTEGRATED PACKET/FLOW MODELING APPROACH

In this section we couple the flow-level ideal GPS model with the packet-level model, into an integrated packet/flow model. Section 3.1 gives the basic steps of our modeling approach and further refinements are presented in Section 3.2.

### 3.1. Basic steps

The first step is to use the packet-level model described in Section 2.2 to determine the aggregate throughput  $t_n$  and the packet loss probability  $p_n$  for the case that a fixed

number of  $n$  flows are active. In particular, this is done by solving the fixed point equation (3). Lost packets trigger the TCP flow control mechanism (i.e., cause TCP to reduce its actual window size) and have to be retransmitted, thus reducing the throughput of the flows. The resulting aggregate flow ‘goodput’ amounts to  $s_n := t_n(1 - p_n)$ .

Next, on the flow level, the system is assumed to behave as a PS model with state dependent service rates  $r_n := s_n$  when  $n$  flows are present. Note that this PS model fits in the framework of GPS models, as discussed in Section 2.1. Hence it still possesses the attractive insensitivity and linearity properties that were described in Section 2.1.

### 3.2. Refinements

**I. Impact of queueing delay.** The original TCP throughput equation (2) assumes that the queueing delay at the bottleneck is negligible compared to the magnitude of the total RTT. However, the effect of queueing delay can be (heuristically) included in the model. The total round trip time RTT consists of a constant part  $\text{RTT}_0$  representing the propagation delays, transmission delays, etc., and a variable part capturing the queueing delay at the bottleneck; the latter part depends on the offered traffic rate  $t_n$ . Thus, RTT can be expressed as

$$\text{RTT}(t_n) = \text{RTT}_0 + d(t_n), \tag{4}$$

where  $d(\cdot)$  is the mean queueing delay in an M/D/1 queue with link speed  $C$  and finite buffer  $K$ . Round trip time  $\text{RTT}(t_n)$ , as follows from (4), can be inserted into the fixed point equation (3). Still, the right hand side decreases in  $t_n$ , ensuring the existence and uniqueness of the fixed point.

**II. Delay correction for initial slow start effect.** We here describe a heuristic to compensate for the initial slow-start phase of a newly arriving TCP flow, cf. [11]. During this phase a possibly considerable number (depending on the average window size) of packets are ‘left unsent’ compared with the ideal behavior of the GPS system; the latter assumes that TCP can start sending at its average speed immediately.

More specifically, the heuristic is as follows. From the steady-state distribution at the flow level (1), the mean number of concurrent flows in the system,  $\mathbb{E}N$ , can be computed, as well as the mean packet arrival rate  $\sum_{n=1}^{\infty} \mathbb{P}(N = n)t_n$ . In the sequel, to make the notation a little lighter, we represent averages by a bar sign above the variable in question, for example  $\bar{t}$ ,  $\bar{N}$ , etc. In equilibrium, the mean window size of a TCP flow,  $\bar{w}$  (in packets), approximately equals  $\bar{w} \approx \text{RTT}(\bar{t}) \cdot \min(r, C/\bar{N})$ , where  $\text{RTT}(\bar{t})$  is given by (4), and  $r \cdot \text{RTT}(\bar{t})$  corresponds to the maximum window size limitation imposed by the access link rate.

Now consider the initial slow start of a newly arriving TCP flow. The time (in RTTs) it takes for the window to grow up to  $\bar{w}$  equals  $k = \lfloor \log_2 \bar{w} \rfloor$ . During this time, compared with an idealized source having the full window size  $\bar{w}$  immediately (as is assumed in the GPS models above), the number of unsent packets equals

$$b = \sum_{i=0}^k (\bar{w} - 2^i) = (k + 1)\bar{w} - (2^{k+1} - 1) = (\lfloor \log_2 \bar{w} \rfloor + 1)\bar{w} + 1 - 2^{\lfloor \log_2 \bar{w} \rfloor + 1}.$$

The time to transmit  $b$  packets at the average sending rate  $\bar{w}/\text{RTT}(\bar{t})$  is then approximately given by  $f = b \cdot \text{RTT}(\bar{t})/\bar{w}$ , and this  $f$  is to be added to the mean flow transmission delay obtained from the flow-level model through ‘Little’, i.e.,

$$\bar{D}_{\text{tot}} = \frac{\bar{N}}{\lambda} + f.$$

## 4. NUMERICAL RESULTS

In this section we present numerical results obtained from our model, and compare them against simulation results, which have been produced by using the ns2 simulator version 2.1b8a<sup>2</sup>, and against numerical results from the ideal GPS model. The integrated packet/flow model described in Section 3 (i.e., including the refinements proposed in Section 3.2) will be here referred to as the *TCP-GPS model*. The basic scenario that is considered is a single bottleneck link with capacity  $C$  being shared by a varying number of TCP Reno flows that have a peak rate limitation due to a limited access link rate  $r < C$ . Unless otherwise specified, the parameters that are fixed during the simulations are the following: the bottleneck link rate  $C = 10$  Mbps, mean flow size  $1/\mu = 1000$  packets and packet size equals 1500 bytes. In the examples below, we vary the access rates, round trip times and buffer sizes. In most of the experiments we assume that the flows have the same, deterministic size  $1/\mu$ ; however, to study the insensitivity property, experiments with exponential and truncated Pareto file sizes are also shown. Finally, the TCP throughput equation used in the packet-level model is the one from Kelly [5],

$$t_n = \frac{n}{\text{RTT}} \sqrt{\frac{2(1-p)}{p}}.$$

For small  $p$  the above simplifies to  $t_n \approx n\sqrt{2}/(\text{RTT}\sqrt{p})$ , i.e.,  $\Gamma = \sqrt{2}$  in (3). Numerical experiments (results not shown due to lack of space) with the more complex throughput expression from [10] have revealed that the effect on the results is only marginal.

The simulation results were obtained by making 100 independent replicas of simulation runs, each consisting of a simulation time equalling the time it takes to complete 200 file transfers. To remove the transient effects, 20 file transfer times from the beginning and the end were ignored. In most of the figures below, confidence intervals of the point estimates have also been given and are indicated by bars.

### 4.1. Experiments with varying RTTs, buffer sizes and access rates

First we explore the effect of buffer size and RTT. The access link rate  $r$  has been chosen to be 1 Mbps. To be specific, the following parameter values are explored: buffer sizes  $K = \{10, 50\}$  and  $\text{RTT}_0 = \{40, 200, 400\}$  ms. Results for the average flow transfer delays for  $K = 10$  and  $K = 50$  are given in Figure 1 and Figure 2, respectively.

As can be seen from the figures, the TCP-GPS model is able to produce qualitatively the same results as simulations, i.e., the model nicely captures the impact of the RTTs and buffer sizes on the flow transfer delay. This is in contrast with the ideal GPS model (as was introduced in Section 2.1), where these (packet-level) parameters do not affect the delays.

<sup>2</sup>Scripts available from Pasi Lassila, pasi.lassila@hut.fi

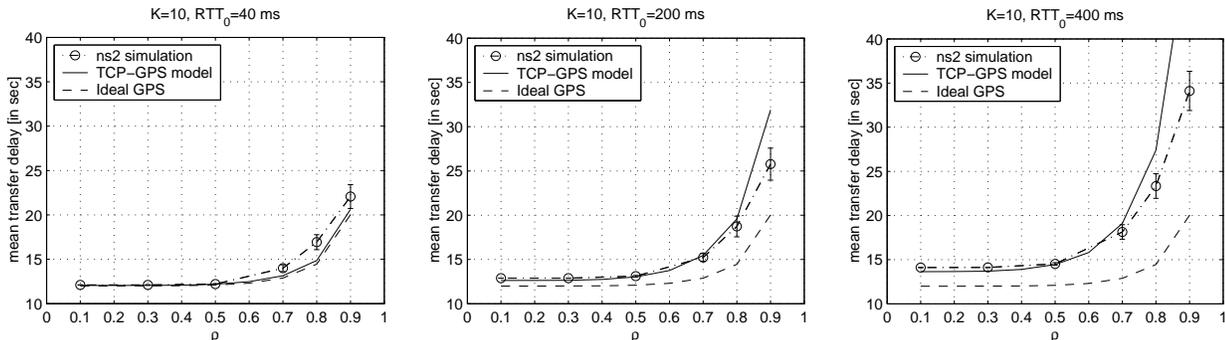


Figure 1. Mean delays for  $K = 10$ , and  $RTT_0 = 40$  ms (left), 200 ms (center), and 400 ms (right).

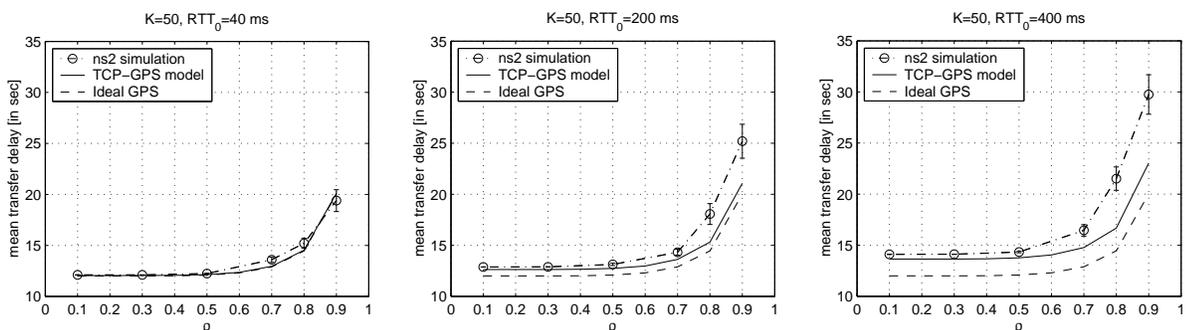


Figure 2. Mean delays for  $K = 50$ , and  $RTT_0 = 40$  ms (left), 200 ms (center), and 400 ms (right).

However, in a quantitative sense, the ns2 accuracy depends on the chosen parameter values. In general, it can be noted that the accuracy of the model is better for the small-buffer cases ( $K = 10$ ) and  $RTT_0$  corresponding to 40 ms and 200 ms. For  $K = 50$  the loss rates in the underlying packet-level queueing model become so small that the TCP equation (3) gives a goodput equalling the link rate  $C$ , i.e., the solution is the same as given by the ideal GPS system. Hence, the difference between the ideal GPS and TCP-GPS results in Figure 2 (for large values of  $RTT_0$ ) is caused by our initial slow start compensation, and not by the ‘TCP-GPS state probabilities’. We also observe that in the simulated system the delays are not affected that much by the buffer size.

To further illustrate the TCP-GPS model, Figure 3 shows also the actual distribution of the number of ongoing flows present in the system for the TCP-GPS model and corresponding ns2 simulations. It can be seen that the distribution of the number of concurrent flows  $N$  matches well with the simulated results. This shows that our approach is an interesting alternative to the procedure described in [4], in which distributions of  $N$  were *a priori* postulated, rather than determined by the model.

Next we illustrate the effect of increasing the access link rate and experiment with setting  $r = 2$  Mbps for  $RTT_0 = \{40, 400\}$  ms and  $K = \{10, 50\}$ . The results are shown

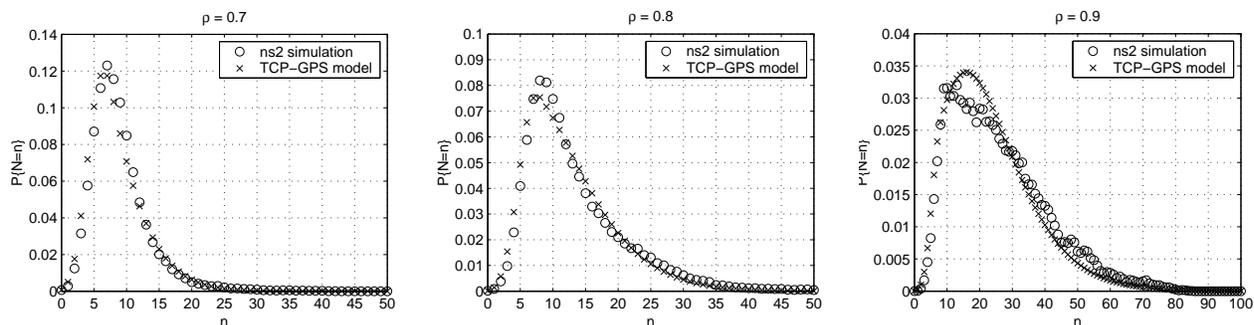


Figure 3. Distribution of the number of flows from simulations and the TCP-GPS model with  $K = 10$  and  $RTT_0 = 200$  ms for  $\rho = 0.7$  (left),  $\rho = 0.8$  (center), and  $\rho = 0.9$  (right).

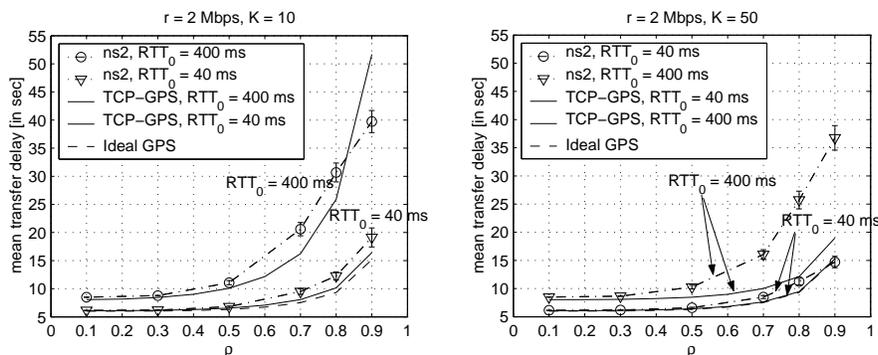


Figure 4. Mean delays for  $K = 10$  (left) and  $K = 50$  (right) with  $r = 2$  Mbps.

in Figure 4. As can be seen from the results, the accuracy is still acceptable, especially for the small-buffer case. However, the simulation results show that, interestingly, in particular cases the mean delays *grow* when increasing the access rate, as opposed to what the TCP-GPS model predicts. A possible explanation for this relates to our Poisson assumption at the packet level. For higher access rates the packet arrival process may become essentially more bursty than Poisson, and thus the estimated packet losses from the M/D/1/K model are too optimistic.

#### 4.2. Experiments with different flow-size distributions

In the previous simulations we assumed a fixed flow size. Our model suggests insensitivity with respect to the flow-size distribution. Here we assess this attractive property of the model by comparing the earlier simulations with constant flow sizes against simulations with exponentially and truncated Pareto distributed flow sizes (with the same mean value). The Pareto distribution has been truncated at 20 000 packets and has shape parameter 1.5. Results are shown in Figure 5 for  $RTT_0 = \{40, 400\}$  ms and  $K = \{10, 50\}$  (confidence intervals have been omitted for clarity). As seen from the results, the mean delays are indeed just modestly affected by the distribution of the flow sizes.

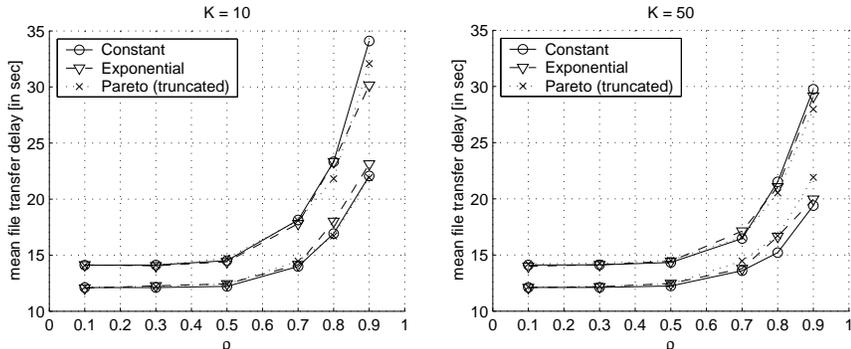


Figure 5. Mean flow transfer delays with constant, exponentially and truncated Pareto distributed flow-sizes for  $K = 10$  (left) and  $K = 50$  (right).

## 5. CONCLUSIONS AND FURTHER RESEARCH

In this paper we have developed a processor sharing type of model for TCP performance analysis, which does not only capture important flow-level effects (i.e., flow arrivals and departures), but takes also into account the impact of packet-level parameters, such as the buffer size and RTT. This integrated packet/flow model can be considered as an enhancement of the ideal GPS flow-level model proposed in other papers.

Comparing the results from our model with simulations, we conclude that in most cases the achieved accuracy is remarkably high. The integrated packet/flow model reflects quite well the impact of packet-level parameters (such as the buffer size and RTT) on the flow transfer delay. This explains why our model outperforms the ideal GPS model, that neglects these packet-level issues. For large buffer sizes, when packet loss probabilities become small, the two models yield similar results. The attractive insensitivity property of the GPS model is maintained; extensive simulations confirm that the average flow transfer delay is indeed hardly affected by the flow size distribution.

The integrated packet/flow model has – as is almost inherent to any modeling approach – its specific weaknesses. In particular, for higher access speeds in combination with large RTTs, the mean flow transfer time is considerably underestimated. We expect that this is due to underestimation of packet loss by the underlying  $M/D/1/K$  model. More precisely, in the case of high access speeds the packet arrival process at the bottleneck link will be quite bursty and the Poisson assumption too optimistic. This packet-level burstiness may also explain a counter-intuitive phenomenon shown by some of our simulations: for some model instances mean flow transfer delays *increase* for higher access speeds.

*Future research:* Our modeling approach relies heavily on the following two assumptions: (i) Poisson packet arrivals at the buffer of the bottleneck link, and (ii) immediate adaptation (after flow arrivals and departures) of the throughput of the TCP sources to a ‘fair share’ of the achievable throughput (i.e.,  $t_n/n$  when  $n$  flows are present). In our opinion, to further enhance our model, the most promising direction is to improve on Assumption (i), i.e., the estimation of the packet arrival process. For that purpose the impact of, e.g., the access speed and RTT variation on the packet arrival process has to

be investigated, as argued above.

Finally, our layered modeling approach can also be used for the following applications. One possible application is the performance of TCP and, e.g., UDP traffic streams multiplexed on a link using a common buffer. Then it is a natural approach to model the aggregate UDP packet arrival process by a Poisson process and to add this to the TCP packet arrival process in the underlying packet-level M/D/1/ $K$  model. Another application refers to the situation where most of the TCP flows are very small (usually referred to as *web mice*, consisting of just a few packets). The (aggregate) packet arrival process of these small flows will hardly be influenced by TCP's flow control, and hence it may be better modelled by an additional (autonomous) Poisson arrival process. The Poisson stream of web mice can then be added to the M/D/1/ $K$  queue.

## REFERENCES

1. K.E. Avrachenkov, U. Ayesta, E. Altman, P. Nain, and C. Barakat, "The effect of router buffer size on the TCP performance", in Proc. of International Seminar on Telecommunication Networks and Teletraffic Theory, St. Petersburg, Russia, Jan 2002.
2. T. Bonald, A. Proutiere, G. Regnie, and J.W. Roberts, "Insensitivity results in statistical bandwidth sharing", in Proc. of ITC-17, Salvador da Bahia, Brazil, Dec 2001.
3. J.W. Cohen, "The multitype phase service network with generalized processor sharing", Acta Informatica 12, 1979.
4. R.J. Gibbens, S.K. Sargood, C. Van Eijl, F.P. Kelly, H. Azmoodeh, R.N. Macfadyen, and N.W. Macfadyen, "Fixed-point models for the end-to-end performance analysis of IP networks", in Proc. of 13th ITC Specialist Seminar: IP Traffic Measurement, Modeling and Management, Monterey, California, USA, Sep 2000.
5. F. Kelly, "Mathematical modeling of the Internet", in Mathematics Unlimited - 2001 and Beyond (Editors B. Engquist and W. Schmid), Springer-Verlag, Berlin, 2001.
6. P. Kuusela, P. Lassila, J. Virtamo, and P. Key, "Modeling RED with idealized TCP sources", in Proc. of IFIP ATM & IP, Budapest, Hungary, Jun 2001.
7. L. Massoulié and J.W. Roberts, "Arguments in favour of admission control for TCP flows", in Proc. of ITC-16, Edinburgh, United Kingdom, Jun 1999.
8. M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm. Computer Communication Review", no. 3, 1997.
9. R. Nunez, H. van den Berg, and M. Mandjes, "Performance evaluation of strategies for integration of elastic and stream traffic", in Proc. of ITC-16, Edinburgh, United Kingdom, Jun 1999.
10. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation", in Proc. of ACM SIGCOMM'98, Vancouver, Canada, Sep 1998.
11. A. Riedl, T. Bauschert, M. Perske, and A. Probst, "Investigation of the M/G/R processor sharing model for dimensioning of IP access networks with elastic traffic", in Proc. of First Polish-German Teletraffic Symposium, Dresden, Germany, Sep 2000.
12. R. Vranken, R.D. van der Mei, R.E. Kooij, and J.L. van den Berg, "Performance of TCP with multiple priority classes", in Proc. of International Seminar on Telecommunication Networks and Teletraffic Theory, St. Petersburg, Russia, Jan 2002.