

# A Multi-level TCP Model with Heterogeneous RTTs

Pasi Lassila<sup>1</sup> and Michel Mandjes<sup>2,3</sup>

<sup>1</sup> Helsinki University of Technology, P.O.Box 3000, FIN-02015-HUT, Finland.

`Pasi.Lassila@hut.fi`

<sup>2</sup> CWI, P.O. Box 94079, 1090 GB, Amsterdam, the Netherlands.

`Michel.Mandjes@cwi.nl`

<sup>3</sup> University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands.

**Abstract.** This paper studies TCP performance focusing on the mean flow transfer delay and the average number of concurrent flows in the system. This is done for a dynamic population of users, rather than a static number of permanent flows. The modeling approach relies on using idealized processor sharing models as a starting point and modifying their properties to take into account some of TCP’s non-ideal behavior. To this end, a model is derived that incorporates the effect of packet losses and RTTs on the goodput, as well as limited sending rates. Also, the unequal sharing of bandwidth between flows with different RTTs can be treated. The delay estimates take into account the initial slow start. Extensive ns2 simulations are used to verify the accuracy of the models.

## 1 Introduction

TCP data transfers account for most of the traffic volume in the Internet. TCP has been designed to support efficient and reliable transmission of data that tolerates variations in the throughput. An important task in TCP performance analysis is to study the dependence of throughput and file transfer delays as functions of the traffic parameters and network parameters. Traditional approaches to modeling TCP performance can be roughly grouped into two classes, which we call (i) flow-level models, and (ii) packet-level models.

Flow-level models are idealized models that include flow level dynamics, but ignore the impact of packet losses and round trip times (RTTs) on TCP performance. So-called processor sharing (PS) queues can be used in this context, see, e.g., [1] and [2]. PS queueing models and a number of their generalizations are attractive due to their insensitivity properties. Packet-level models capture more details of the system (RTTs, buffer size, etc.), but assume a constant number of persistent flows. Here an important result has been the “square-root-p” formula (see, e.g., [3]), which captures the impact of RTT on the throughput, for a given packet loss probability. By noticing that the throughput affects, in return, the packet loss probability, fixed-point models can be used to compute both throughput and loss probability, see, e.g., [4], [5] and [6]. The work in [7]

on modeling the conditional mean file transfer delay is also a packet-level model as no flow-level dynamics are considered.

Work on combined flow/packet-level modeling is done by Gibbens *et al.* [5]. To take into account the fluctuating number of concurrent flows  $N$ , they assume a priori  $N$  to be either Poisson or geometrically distributed. In Lassila *et al.* [8], a model is presented that combines both levels by computing first, at the packet level, the conditional throughput given the number of flows. These are then used as input to the flow level model by using a model that is sometimes called Generalized Processor Sharing (GPS) [9]. Hence the distribution of  $N$  results from the model. Both approaches apply a packet-level queueing model to obtain the throughput (as a fixed point). In Massoulié and Roberts [2], a GPS model is utilized, as well, but no queueing model is used to obtain the packet loss probabilities. Bu and Towsley [10] also study a single congested link and use the idealized unequal processor sharing model in [11] to provide a model for the mean flow transfer delays. In Ayesta *et al.* [4], a model for short TCP flows with stochastic flow arrivals has been given in, where the conditional mean-delay model is parameterized by the packet-level estimate for packet loss. To estimate the mean number of flows in the system, an M/G/ $\infty$  approximation is used.

In this paper we develop a combined flow/packet-level model for studying TCP performance under user-heterogeneity, in the sense of users having different RTTs. We first consider a group of TCP users with identical RTTs, sharing a single congested link equipped with a finite buffer, and then generalize the model to heterogeneous RTT user groups with an access-rate limitation. Our contribution is that our model, while being tractable, covers all above features, whereas previous models included them only partially. The model in [10] considers user heterogeneity, but not the impact of limited access rates, nor finite buffers. On the other hand, the model in [4] considers buffer sizes and limited access rates, but not the impact of bandwidth sharing. The GPS model in [8] captures finite buffers and limited access rates, but can not be easily extended to heterogeneous RTTs. Our approach is based on deriving a system of differential equations representing the expected change per time unit in the packet sending rate (throughput) and the number of flows in the system. Thus, it is a generalization of the model in [6], where the number of flows is constant. Here we add a flow-level equation to the system. From the model we are able to obtain the mean delay and mean number of flows in the system. To increase the accuracy of the mean-delay estimates, additional heuristics are given to take into account the effect of initial slow starts. The models are validated through ns2 simulations.

## 2 TCP Model with Stochastic Flow Arrivals

We model the ideal behavior of stochastically arriving Reno-type TCP flows in the so-called congestion avoidance phase sharing the capacity of a single bottleneck link. Thus, timeouts are ignored and it is assumed that the sending rate of a TCP source increases linearly as long as acknowledgements are received and that the rate is halved for each packet loss. To this end a model is derived which

consists of two parts: a flow-level model and a window-level model. At the packet level, we make a stationarity assumption and use a finite-buffer M/G/1 queue model (finite M/M/1 and M/D/1 models are used in the numerical results). The model is first derived for TCP users with the same RTT (homogeneous users) and then the model is generalized to the case of heterogeneous RTTs.

## 2.1 Homogeneous TCP Users

Consider TCP flows sharing a link with capacity  $C$  (pkts/sec) and buffer size  $K$  packets. Flows arrive according to a Poisson process with rate  $\nu$  and the file lengths are assumed to be exponentially distributed with mean  $1/\mu$  packets. The aggregate mean packet sending rate of the TCP population is denoted by  $\lambda(t)$  at time  $t$ . All flows are assumed to have the same RTT,  $R(\cdot)$ ,

$$R(t) = R_0 + d(t) ,$$

where  $R_0$  represents the constant packet transmission and propagation delays and  $d(t)$  is the mean delay in an M/G/1/K buffer with arrival rate  $\lambda(t)$ . The mean window size of the TCP population is denoted by  $\bar{W}(t)$ , and thus, the mean sending rate is given by  $\lambda(t) = \bar{W}(t)/R(t)$ . Correspondingly, the mean sending rate of a single TCP flow, that we give index  $i$ , is  $\lambda_i(t) = \bar{w}_i(t)/R(t)$ , where  $\bar{w}_i(t)$  is the mean window size of flow  $i$ .

**Flow-level equation:** Consider a small time interval  $\Delta t$ . At the flow level, during  $\Delta t$  either a flow arrives or a flow departs. As flows arrive at rate  $\nu$ , a flow arrival occurs with probability  $\nu\Delta t$ . Given that there are flows present in the system, a single flow in the system sends at mean rate  $\lambda_i(t)$ , but the goodput is  $\lambda_i(t)(1 - P(t))$ , where  $P(t)$  is the packet loss probability. Thus, the probability of flow  $i$  completing its transfer during  $\Delta t$  equals  $\mu\lambda_i(t)(1 - P(t))\Delta t$ . Assuming that the flows are independent, the probability that any flow finishes is  $\mu\lambda(t)(1 - P(t))\Delta t$ , where  $\lambda(t) = \sum_i \lambda_i(t)$ . The change in the mean number of flows,  $\bar{N}(t)$ , during  $\Delta t$  can be expressed as

$$\Delta\bar{N}(t) = \nu\Delta t - \mu\lambda(t)(1 - P(t))(1 - \pi_0(t))\Delta t ,$$

where  $(1 - \pi_0(t))$  is the probability that there are flows present in the system. Letting  $\Delta t \rightarrow 0$  we obtain

$$\frac{d\bar{N}(t)}{dt} = \nu - \mu\lambda(t)(1 - P(t))(1 - \pi_0(t)) . \quad (1)$$

The dynamics of  $\pi_0(t)$  are unknown, but at the flow level, we assume that  $N(t)$ , the process of the number of flows present in the system at time  $t$ , behaves as in an ordinary PS queue. Even then we do not have an exact expression for the dynamics of  $N(t)$ , but we can use a similar quasi stationarity approximation as is done at the queue level in [6]. In equilibrium,  $\pi_0$  is a function of the load  $\rho$ ,  $\pi_0 = 1 - \rho$  and  $\bar{N}$  is also a function of  $\rho$ ,  $\bar{N} = \rho/(1 - \rho)$ . Eliminating  $\rho$  from these gives  $1 - \pi_0 = \bar{N}/(1 + \bar{N})$ . Thus, the differential equation (1) becomes

$$\frac{d\bar{N}(t)}{dt} = \nu - \mu\lambda(t)(1 - P(t))\frac{\bar{N}(t)}{1 + \bar{N}(t)} . \quad (2)$$

Note that in the above  $\lambda(t)(1 - P(t))$  represents the aggregate goodput of the flows, conditional on the event that there is a positive number of flows in the system. In an ordinary PS system this goodput is equal to  $C$ , the bandwidth of the bottleneck, implying that the flows can utilize the capacity fully without packet losses. In reality, TCP only approximates this ideal behavior, and packet losses and RTTs affect the goodput of the flows. To capture this effect, we next derive the equation describing the change in the expected value of the conditional mean sending rate  $\lambda(t)$ . In effect, the idea here is to compensate for the non-ideal performance of TCP compared to PS by assuming the system to behave as a PS system only with a smaller goodput than in the ideal PS system.

**Window-level equation:** In [6], for a fixed number, say  $n$ , of TCP flows, the change in  $\Delta t$  in  $\lambda(t)$  has been shown under some approximations to be

$$\Delta\lambda(t) = \frac{\Delta\bar{W}(t)}{R(t)} = \frac{n}{R(t)} \left( (1 - P(t)) \frac{1}{\bar{w}_i(t)} - P(t) \frac{\bar{w}_i(t)}{2} \right) \lambda_i(t) \Delta t, \quad (3)$$

where  $\lambda_i(t)\Delta t$  is the probability of a packet arrival from flow  $i$  and the term in brackets represents the change to the aggregate window size if a packet is accepted ( $1/\bar{w}_i$  term) or lost ( $\bar{w}_i/2$  term). In the setting of the present paper,  $n$  is a random variable which evolves according to a stochastic process  $N(t)$ . However, assuming that (3) holds approximately for any  $n$ , averaging it over the distribution of  $N(t)$  gives

$$\Delta\lambda(t) = \frac{\bar{N}(t)}{R(t)} \left( (1 - P(t)) \frac{1}{\bar{w}_i(t)} - P(t) \frac{\bar{w}_i(t)}{2} \right) \lambda_i(t) \Delta t.$$

Doing so means that the arrival and departure of flows are not modelled at the window level. Instead, we assume the time scale of changes in  $N(t)$  to be much slower than that of the changes to the window sizes, i.e., that the file transmission times are long (cf. the separation of time scales principle). Thus, upon a change in  $N(t)$ , the window of a new flow quickly reaches the new stationary value and the other flows also adapt to the situation quickly (similarly for flow departures). Then, by noting that  $\lambda_i(t) = \bar{w}_i(t)/R(t)$  and that  $\lambda_i(t) = \lambda(t)/\bar{N}(t)$ , and letting  $\Delta t \rightarrow 0$ , we obtain

$$\frac{d\lambda(t)}{dt} = (1 - P(t)) \frac{\bar{N}(t)}{R(t)^2} - P(t) \frac{\lambda(t)^2}{2\bar{N}(t)}. \quad (4)$$

At the packet level in the queue, it is assumed that the window level always observes packet losses and queuing delays resulting from a stationary M/G/1/K system with arrival rate  $\lambda(t)$ .

**Complete model and steady-state solution:** Combining (2) and (4), the complete model is given by

$$\begin{cases} \frac{d\bar{N}(t)}{dt} = \nu - \mu\lambda(t)(1 - P(t)) \frac{\bar{N}(t)}{1 + \bar{N}(t)}, \\ \frac{d\lambda(t)}{dt} = (1 - P(t)) \frac{\bar{N}(t)}{R(t)^2} - P(t) \frac{\lambda(t)^2}{2\bar{N}(t)}, \end{cases} \quad (5)$$

where  $R(t) = R_0 + d(t)$ ,  $d(t)$  is the mean delay in an M/G/1/K system with arrival rate  $\lambda(t)$  and  $P(t)$  equals the loss probability in an M/G/1/K system with arrival rate  $\lambda(t)$ .

The steady-state solution of (5) is obtained by setting its right hand side equal to zero. From the second equation for  $\lambda(t)$ , we get  $\bar{N} = \alpha R \lambda$ , where  $\alpha = \sqrt{P/(2(1-P))}$ . Inserting this in the first equation results in a second order equation, for which the only positive solution equals

$$\lambda = \frac{\nu}{2\mu(1-P)} \left( 1 + \sqrt{1 + \frac{4\mu(1-P)}{\nu\alpha R}} \right). \quad (6)$$

Note that to solve the above one needs to solve a fixed point equation since by (5) both  $P$  and  $R$  depend on the value of  $\lambda$ . Although confirmed by extensive numerical experiments, we have not succeeded in formally proving uniqueness of the fixed point.

**The mean-delay model:** The above model would already enable us to model the mean delay by simply using Little, i.e., the mean delay equals  $\bar{N}/\nu$ . However, this is not a very accurate model of reality as it assumes that the mean goodput rate of a single flow  $\lambda(1-P)/\bar{N}$  is available instantly, whereas in reality there is a certain time how long it takes for the TCP's sending rate (window size) to grow up to the estimated steady-state rate.

Our basic idea is to approximate the total mean delay of a file transfer,  $\bar{D}_{tot}$ , by taking into account how much of the file is sent during the time it takes to reach the estimated steady-state goodput rate (initial slow start),  $\bar{D}_{ss}$ , and the remaining file size is then sent at the steady-state goodput rate,  $\bar{D}_{eq}$ , i.e.,  $\bar{D}_{tot} = \bar{D}_{ss} + \bar{D}_{eq}$ . Observe that often it can also happen that the file size is so small and/or the bandwidth-delay product is so large that the predicted goodput rate is never reached, and the whole file is transmitted during the initial slow start. From our above model, the following performance measures are obtained: packet loss probability  $P$ , total goodput rate  $\lambda(1-P)$ , and mean RTT including the queuing delay  $R$ . The mean goodput rate equals approximately  $\lambda(1-P)/\bar{N}$ , except for cases where the mean number of flows in the system is so low (i.e., load is low) that  $\lambda(1-P)/\bar{N} > C$ , in which case we simply approximate that the goodput rate per flow equals  $C$ . Thus, the goodput rate of the TCP flows,  $\bar{r}$ , equals  $\bar{r} = \min(C, \lambda(1-P)/\bar{N})$ , which corresponds to a window size  $\bar{w} = \bar{r} \cdot R$ .

On the other hand, the mean time  $m$  (expressed in the number of RTTs) to send out a file of size  $1/\mu$  assuming that the entire file is sent during slow start, is obtained from  $\sum_{i=0}^m 2^i = 1/\mu$  yielding  $m = \lceil \log_2(1/\mu - 1) \rceil - 1$ . Note that the numbering of rounds starts from index 0 (actual number of rounds equals  $m+1$ ). If  $2^m \leq \bar{w}$  the file is sent during the slow start before the system reaches equilibrium, i.e.,  $\bar{D}_{ss} = (m+1)R$ . If  $2^m > \bar{w}$ , the flow reaches equilibrium in  $a = \lfloor \log_2 \bar{w} \rfloor$  rounds (indexing starts from 0 again) during which  $(1-2^{a+1})/(1-2) = 2^{a+1} - 1$  packets are sent. The time to transmit the file equals the sum of the mean delay from slow start,  $\bar{D}_{ss} = (a+1)R$ , and the equilibrium delay  $\bar{D}_{eq} = (1/\mu - 2^{a+1} + 1)/\bar{r}$  (time it takes to send the remaining packets at the equilibrium goodput rate). Putting the above pieces together gives us the following model

for the mean transfer delay of files of size  $1/\mu$ ,

$$\bar{D}_{tot} = \begin{cases} \bar{D}_{ss} = (m+1)R, & \text{if } 2^m \leq \bar{w}, \\ \bar{D}_{ss} + \bar{D}_{eq} = (a+1)R + (1/\mu - 2^{a+1} + 1)/\bar{r}, & \text{otherwise,} \end{cases} \quad (7)$$

where  $m = \lceil \log_2(1/\mu - 1) \rceil - 1$  and  $a = \lfloor \log_2 \bar{w} \rfloor$ .

## 2.2 Heterogeneous TCP Users and Limited Access Rates

Here we present heuristics to extend the model to the case of  $M$  heterogeneous user groups, each with their own flow arrival rate  $\nu_k$  and constant link delays  $R_{0,k}$ , for  $k = 1, \dots, M$ . Each TCP user group may also have a limitation in their sending rate,  $\lambda_{\max}^k$ , caused, e.g., by an access link. The GPS model [9] does not apply under user heterogeneity and hence the approach in [8] can not be used. An idealized model for unequal bandwidth sharing is the so-called DPS model [11] (and applied in [10]). However, in its full generality it allows the mean file size to be different among the user groups and this model cannot be easily modified to take into account the effect of packet losses on goodput. Here we utilize the properties of the DPS model under the natural assumption of a common mean file length for all user groups and aim to give a simple approximate model that captures both the effect of a limited access rate, packet losses and different RTTs.

To model the above, observe that when the load of the system is low enough such that the number of flows in the system rarely exceeds the limit at which the bottleneck fills up, the system behaves as an M/G/ $\infty$  system where the flow is only constrained by its access link rate and the behavior of TCP slow start. When the load is high enough to fill the bottleneck link, the system becomes a processor sharing system where the flows share the bandwidth. In this case, the total arrival rate of flows equals  $\sum_k \nu_k$  and the aggregate goodput is  $\sum_k \lambda_k(t)(1 - P(t))$ , where  $\lambda_k(t)$  is the mean sending rate of population  $k$  at time  $t$ . Now it is easy to see that for any work conserving service discipline (such as DPS), assuming exponential file lengths with a common mean  $1/\mu$  for all classes, the total number of flows in the system is a Markov process with the same properties as the number of users in an ordinary PS system. Hence, we can use the same model for the time evolution of the mean total number of flows in the system  $\bar{N}(t)$  as earlier,

$$\frac{d\bar{N}(t)}{dt} = \sum_k \nu_k - \mu \sum_k \lambda_k(t)(1 - P(t)) \frac{\bar{N}(t)}{1 + \bar{N}(t)}. \quad (8)$$

For each TCP user group, the sending rate  $\lambda_k(t)$  is determined by

$$\frac{d\lambda_k(t)}{dt} = (1 - P(t)) \frac{\bar{N}_k(t)}{R_k(t)^2} - P(t) \frac{\lambda_k(t)^2}{2\bar{N}_k(t)}, \quad (9)$$

where  $R_k(t) = R_{0,k} + d(t)$  and  $N_k(t)$  is the mean number of population  $k$  flows.

Given (8) and (9), we need a model for  $\bar{N}_k(t)$  that captures the unequal sharing of the bottleneck bandwidth among heterogeneous users. In our system, the

load caused by population  $k$  equals  $\nu_k/(C \cdot \mu)$ . However, according to our model, each population obtains a goodput equalling  $\lambda_k(t)(1 - P(t))$ , which includes the effects of packet loss and RTT. Thus, the *effective* load caused by population  $k$  is given by  $\nu_k/(\mu \cdot \lambda_k(t)(1 - P(t)))$ , and we approximate  $\bar{N}_k(t)$  by dividing  $\bar{N}(t)$  in proportion to the effective load of each class,

$$\bar{N}_k(t) = \frac{\nu_k/(\mu \cdot \lambda_k(t)(1 - P(t)))}{\sum_k \nu_k/(\mu \cdot \lambda_k(t)(1 - P(t)))} \bar{N}(t) = \frac{\nu_k/\lambda_k(t)}{\sum_k \nu_k/\lambda_k(t)} \bar{N}(t). \quad (10)$$

**Complete model and steady-state solution:** Combining (8) and (9), the complete model is given by

$$\begin{cases} \frac{d\bar{N}(t)}{dt} = \sum_k \nu_k - \mu_k \sum_k \lambda_k(t)(1 - P(t)) \frac{\bar{N}(t)}{1 + \bar{N}(t)}, \\ \frac{d\lambda_k(t)}{dt} = (1 - P(t)) \frac{\bar{N}_k(t)}{R_k(t)^2} - P(t) \frac{\lambda_k(t)^2}{2\bar{N}_k(t)}, \quad k = 1, \dots, M, \end{cases} \quad (11)$$

where  $\bar{N}_k(t)$  is given by (10).

The steady-state values are again obtained by setting the right hand sides of (11) equal to zero. By solving  $\bar{N}$  from the first equation as a function of  $\lambda_k$ , and solving from the rate equation for  $\lambda_k$ , the steady-state solution for  $\lambda_k$  can be expressed in the form

$$\lambda_k = N_k(\lambda_1, \lambda_2, \dots) \cdot \sqrt{\frac{2(1 - P(\sum_k \lambda_k))}{P(\sum_k \lambda_k)}} \cdot \frac{1}{R(\sum_k \lambda_k)},$$

i.e., in the form of a fixed point equation. To prove the uniqueness of the fixed point is not easy but numerical experiments indicate a unique solution exists.

**The mean-delay model:** The model for the mean delay developed in the previous section can be used in this case with a simple modification. The model above provides estimates of the following performance measures: packet loss probability  $P$ , total goodput rate  $\lambda_k(1 - P)$  of each population, and mean RTT including the queuing delay  $R_k$ . At the equilibrium, the goodput rate of a single flow is limited either by its access link rate,  $\lambda_k^{\max}$ , or, if the mean number of flows in the system is high enough, by the sharing of the bottleneck link. Thus, the goodput rate of the TCP flows of population  $k$ ,  $\bar{r}_k$ , equals  $\bar{r}_k = \min(\lambda_k^{\max}, \lambda_k(1 - P)/\bar{N}_k)$ . With this modification the rest of the model for the file transfer delay is the same as given by (7).

**The mean number of flows in the system:** Here we can use similar ideas as in [4]. In the case, where the bottleneck load is high enough, the model above provides an estimate of the mean number of flows,  $\bar{N}_k$ . However, when the load is below the threshold for sharing, the system operates as an M/G/ $\infty$  system with arrival rate equal to  $\nu_k$  and mean service time equal to  $\bar{D}_{tot,k}$ , as given by (7). In an M/G/ $\infty$  system, the distribution of the number of flows obeys a Poisson distribution with parameter  $\nu_k \bar{D}_{tot,k}$ , and the mean also equals  $\nu_k \bar{D}_{tot,k}$ . Thus, the mean number of flows is obtained from  $\bar{N}_{k,final} = \max(\bar{N}_k, \nu_k \bar{D}_{tot,k})$ .

### 3 Numerical Results

Here we present numerical results obtained from our models, and compare them against simulation results, which have been produced using the ns2 simulator version 2.1b9a. We experiment with different TCP variants (TCP Reno and TCP SACK), buffer sizes, RTTs and flow size distributions. In the simulations, the packet size for all TCP sources is 500 bytes, and the mean file size is 500 packets. We also experiment with different queuing models, namely M/M/1/K and M/D/1/K, which are used as part of our packet-level model.

#### 3.1 Examples with Homogeneous TCP Users

We first consider tests with TCPs having identical RTTs and illustrate the impact of the access rate limitation and the scaling of bottleneck capacity on the performance. By introducing an access rate limit on the flows the packet arrivals can be made less bursty, hence making our Poisson assumption more plausible. By scaling the bottleneck capacity our system becomes more like an M/G/∞ system with constant mean delays and linearly increasing mean number of flows. This is illustrated in Figure 1 for the mean delays and in Figure 2 for the mean number of flows. In all figures, results are shown as a function of the load of the bottleneck link. Each figure has two sets of curves corresponding to results for two different RTTs ( $R_0 = 20$  ms and  $R_0 = 200$  ms) to assess the effect of the bandwidth delay product. In each set of curves, solid lines represent solutions to our analytical models and dashed lines are simulation results.

In Figure 1 (left), the bottleneck capacity  $C = 10$  Mbps, the access links of the flows have the same capacity,  $\lambda^{\max} = 10$  Mbps, and the buffer size  $K = 10$ . To evaluate the effect of the TCP variant, results for TCP Reno and TCP SACK sources are shown. The system is, in a sense, a processor sharing system where any flow can utilize the full link capacity. This produces rather bursty packet arrivals and to compensate for this we have used the M/M/1/K model as our packet level model (as opposed to M/D/1/K, which would be the more ‘realistic’ model). As seen from the results, TCP SACK is able to avoid time outs more effectively than TCP Reno (and thus has lower delays), especially at low loads. Thus, in the following we only use TCP SACK sources. Results for  $R_0 = 200$  ms

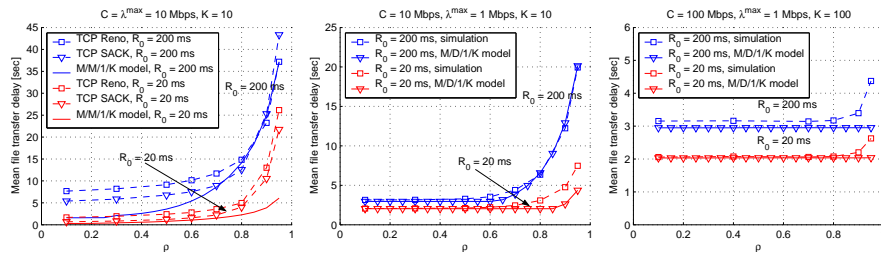
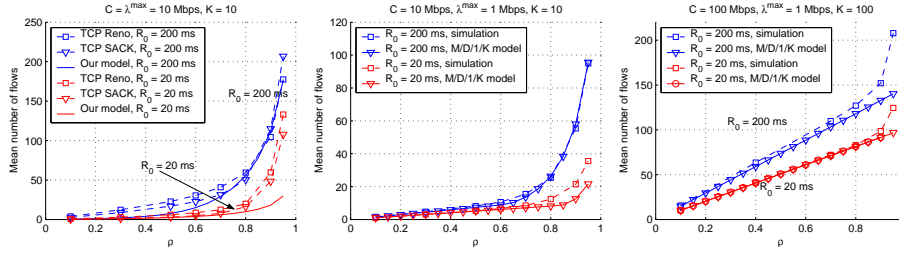


Fig. 1. Mean file transfer delays for three different scenarios.





**Fig. 2.** Mean number of flows for three different scenarios.

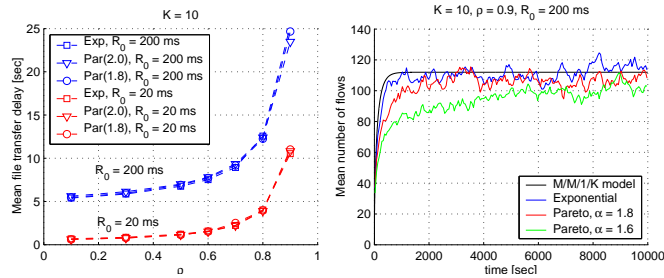
are acceptable but for  $R_0 = 20$  ms the model underestimates the delays. One reason for the inaccuracy is the very bursty nature of the packet arrivals.

Next we introduce an access rate limitation to smoothen the packet arrival process, and hence our Poisson assumption at the packet level should be more appropriate. As TCP actually sends constant size packets, the queue model is M/D/1/K (using M/M/1/K overestimates the packet losses). The result can be seen in Figure 1 (center), where  $C = 10$  Mbps, access rate  $\lambda^{\max} = 1$  Mbps and  $K = 10$ . In the figure it can be seen how the results become more accurate. Finally, we scale the bottleneck capacity to  $C = 100$  Mbps and also increase the buffer size to  $K = 100$  in an attempt to have the system behave as an M/G/ $\infty$  system with constant mean delays. The result is shown in Figure 1 (right). It can be seen that our model, indeed, predicts that the mean delays stay constant but in the high load region the delays in the simulated system are not quite constant and our model underestimates the simulation results somewhat.

Figure 2 contains the results for the mean number of flows in the system for the three cases described above. The accuracy of the results is similar to those of the mean delays. The right figure nicely shows how the system approximates the M/G/ $\infty$  system with linearly increasing mean number of flows.

In general the accuracy depends on the numerous parameters of the system, but due to lack of space a systematic evaluation of the accuracy of the model can not be presented. Shortly, the dependencies are such that smaller buffer sizes give more accurate results; for larger buffers the model underestimates the loss probabilities. For smaller file sizes, our model does not estimate the available goodput for flows accurately, as our assumptions on the time scale decomposition between the rate adaptation time and flow interarrival times does not hold. Moreover, our TCP model does not take into account timeouts, which are more important the smaller the file sizes are.

**System dynamics and effect of different file size distributions:** Here we explore the insensitivity of the steady-state solution and the effect of the distribution on the dynamics. The main emphasis here is on the results concerning system dynamics under different distributions. Similar results on the insensitivity of the steady state have appeared elsewhere in the literature, see, e.g., [8], and they are shown here mainly for the sake of completeness. We study first the effect of different file size distributions on the mean file transfer delays.



**Fig. 3.** Mean steady-state file transfer delays for  $R_0 = \{20, 200\}$  ms (left) and the length of the transient period for different distributions (right).

The distributions that are used are: exponential, Pareto with shape parameter  $\alpha = 2.0$  and  $\alpha = 1.8$ . In the simulation, the constant parameters are:  $K = 10$  and  $C = \lambda^{\max} = 10$  Mbps. The varied parameters are the bottleneck link load and  $R_0 = \{20, 200\}$  ms. As can be seen in Figure 3 (left), the steady-state mean file transfer delays are hardly affected by the distribution. In Figure 3 (right) the accuracy of our dynamical model is illustrated for exponential file lengths. Additionally, we show the impact of the type of the distribution on the time to reach stationarity. In the figure, the transient evolution for mean number of flows in the system is plotted in a system where  $R_0 = 200$  ms,  $K = 10$  and bottleneck load  $\rho = 0.9$ . Three file size distributions are considered: exponential and Pareto with shape parameters 1.8 and 1.6. The averages have been obtained by averaging over 50 sample paths for each case. As can be seen, the dynamics of the simulation with exponential file sizes nicely match the results of our analytical model. Regarding the effect of the distribution on the dynamics, for Pareto with shape parameter 1.8 the transient time is still manageable, though longer than for exponential. However, for Pareto with shape parameter 1.6 the transient to reach stationarity is much longer than for exponential.

### 3.2 Experiments with Heterogeneous RTTs

Next we experiment with four TCP populations with different RTTs that share a single bottleneck. Only results for the mean delays are presented due to lack of space (results for the mean number of flows are similar in accuracy). The bottleneck link bandwidth  $C = 10$  Mbps and each TCP source has an access rate  $\lambda^{\max} = 1$  Mbps, and the link delays were chosen such that  $R_{0,i} = \{30, 70, 150, 230\}$  ms. The bottleneck buffer size is either  $K = 10$  or  $K = 100$  and for the analytical results the queue model is M/D/1/K. To explore the impact of the distribution, the file length distributions were exponential and Pareto with shape parameter 2.0. The results for the mean delays are given in Figure 4. We can observe a rather good match between the simulation and analytical results, especially for the small buffer cases. For larger buffers the analytical model gives results that somewhat underestimate the simulated results. Note that the distribution type does not greatly affect the mean delays.

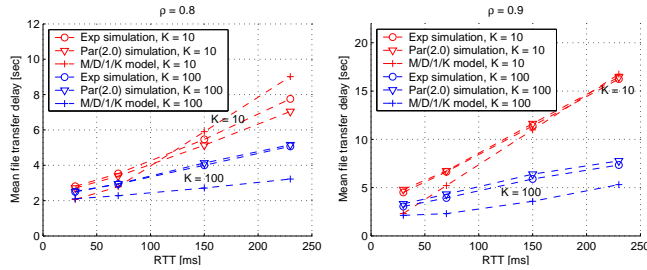


Fig. 4. Mean transfer delays with different RTTs for  $\rho = 0.8$  (left) and  $\rho = 0.9$  (right).

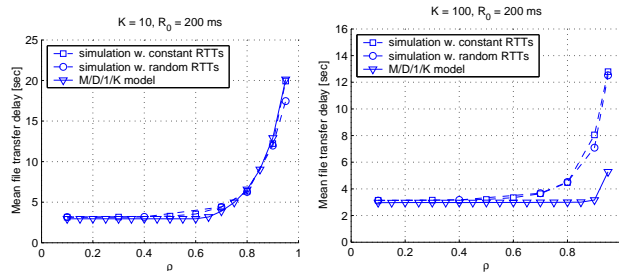


Fig. 5. Mean transfer delays with random RTTs for  $K = 10$  (left) and  $K = 100$  (right).

**Tests with random RTTs:** Here we present results from experiments where the delay on access link of each flow is drawn from a distribution separately for each new flow. We consider the same scenario as earlier with access rates being equal to 1 Mbps and the bottleneck link rate 10 Mbps. The bottleneck link is set to have a constant two-way delay of 10 ms. The two-way delay of the access link is drawn independently for each file transfer from a uniform distribution in the range  $[0, 390]$  ms. Hence, the mean RTT without queuing delay equals 200 ms and the variation in the RTTs of flows is significant. Two different buffer sizes are studied,  $K = 10$  and  $K = 100$  packets, respectively. We compare the results against simulation results with a constant two-way delay of 200 ms and the results from our model with the M/D/1/K queuing model. The results are shown as a function of the bottleneck load in Figure 5 for the mean delays. As can be seen, perhaps even slightly surprisingly, the means are not really affected by the randomness of the access link delays (variability did increase, though). This suggests that from the point of view of just modeling overall mean delays, a model accounting just for the mean RTT seems enough.

## 4 Conclusions

This paper studies TCP performance, with a specific focus on the mean flow transfer delay and the average number of concurrent flows in the system. This is done for a dynamic population of users, rather than the situation of a static

number of permanent flows. The modeling approach relies on using idealized PS models as a starting point and modifying their properties accordingly to take into account some of TCP's non-ideal behavior. A model has been derived that can incorporate the effect of packet losses and RTTs on the obtained goodput, as well as limited sending rates. Also, the unequal sharing of bandwidth between flows with different RTTs can be treated. The delay estimates additionally consider the impact of the initial slow start. Ns2 simulations have been used to verify the accuracy of the models. In general, the accuracy is better for systems where the ratio of the access link rate to the bottleneck link rate is relatively small, which corresponds to a higher degree of multiplexing on the bottleneck. It is also under such circumstances that the assumption of Poisson arrivals (at the packet level) may be assumed to be more applicable. However, the accuracy is in general dependent on the parameters; small buffer sizes usually give more accurate results. Future research topics include the derivation of more accurate packet-level models, modeling the effect of Web mice that share the capacity with longer flows, and extending the models to a multi-hop context.

## References

1. Bonald, T., Proutiere, A., Régnie, G., Roberts, J.: Insensitivity results in statistical bandwidth sharing. In: Proc. of 17th International Teletraffic Congress, Bahia da Salvador, Brazil (2001) 125–136
2. Massoulié, L., Roberts, J.W.: Arguments in favour of admission control for TCP flows. In: Proc. of 16th International Teletraffic Congress, Edinburgh, UK (1999) 33–44
3. Mathis, M., Semke, J., Mahdavi, J., Ott, T.: The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review* **27** (1997) 67–82
4. Ayesta, U., Avrachenkov, K., Altman, E., Barakat, C., Dube, P.: Multilevel approach for modeling short TCP sessions. In: Proc. of 18th International Teletraffic Congress, Berlin, Germany (2003) 661–670
5. Gibbens, R.J., Sargood, S.K., Eijl, C.V., Azmoodeh, F.P.K.H., Macfadyen, R.N., Macfadyen, N.W.: Fixed-point models for the end-to-end performance analysis of IP networks. In: Proc. of 13th ITC Specialist Seminar, Monterey, CA, USA (2000)
6. Kuusela, P., Lassila, P., Virtamo, J., Key, P.: Modeling RED with idealized TCP sources. In: Proc. of IFIP Conference on Performance Modeling and Evaluation of ATM & IP networks, Budapest, Hungary (2001) 155–166
7. Cardwell, N., Savage, S., Anderson, T.: Modeling TCP latency. In: Proc. of IN-FOCOM 2000, Tel Aviv, Israel (2000) 1742–1751
8. Lassila, P., van den Berg, H., Mandjes, M., Kooij, R.: An integrated packet/flow model for TCP performance analysis. In: Proc. of 18th International Teletraffic Congress, Berlin, Germany (2003) 651–660
9. Cohen, J.W.: The multitype phase service network with generalized processor sharing. *Acta Informatica* **12** (1979) 245–284
10. Bu, T., Towsley, D.: Fixed point approximations for TCP behavior in an AQM network. In: Proc. of SIGMETRICS 2001, Cambridge, MA, USA (2001) 216–225
11. Fayolle, G., Mitrani, I., Iasnogorodski, R.: Sharing a processor among many job classes. *Journal of the ACM* **27** (1980) 519–532