

# A multiple time-scale model for TCP bandwidth sharing under user heterogeneity

Dirk Abendroth<sup>1 2</sup>, Hans van den Berg<sup>2 3</sup>, Michel Mandjes<sup>2 4</sup>

<sup>1</sup> Technical University Hamburg-Harburg, Denickestr. 17, 21073 Hamburg, Germany,

<sup>2</sup> University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands,

<sup>3</sup> TNO Telecom, Brasserieplein 2, P.O. Box 5050, 2600 GB Delft, The Netherlands,

<sup>4</sup> CWI, Kruislaan 413, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

**Abstract.** Building on the vast body of existing TCP models, we develop a novel versatile model that explicitly captures user heterogeneity, and takes into consideration dynamics at both the packet level and the flow level. It is described how the resulting multiple time-scale model can be numerically evaluated. Validation is done by using NS2 simulations as a benchmark. In extensive numerical experiments, we study the impact of heterogeneity in the round-trip times on user-level characteristics such as throughputs and flow transmission times, thus quantifying the resulting bias. In particular, we investigate to what extent this bias is affected by the networks' 'packet-level parameters', such as buffer sizes. We conclude by extending the single-link model in a straightforward way to the general network setting.

**Key words:** TCP, user heterogeneity, performance, throughput, round-trip times, packet level, flow level

## 1 Introduction

Enabled by enhanced access technologies such as 'fiber to the home', ADSL, wireless LANs or UMTS, the number of users with high-speed access to the Internet is increasing rapidly. At the same time, more and more Internet applications require some sort of minimum quality-of-service (QoS), expressed in terms of delay, throughput, etc. With most of the data transfers relying on TCP as underlying end-to-end transport protocol, the performance of TCP-controlled networks has become a prominent theme in networking research.

TCP has been designed to support efficient and reliable transmission of elastic data flows, tolerating some variations in the throughput. In particular, based on implicit information about the level of network congestion (round-trip time, packet loss) TCP increases or decreases the sending rate in order (to attempt) to provide a fair share of the network resources to all users. However, it is clear that heterogeneous user behavior might lead to asymmetries in the experienced performance. For instance, one may wonder to what extent it pays off to have a higher access rate or a shorter round-trip time than the other TCP flows. It is this relation between user heterogeneity and flow-level performance (throughput, flow transfer times) that is studied in this paper.

A study on the impact of user heterogeneity could be done relying exclusively on simulation tools (like the NS2 [13] TCP simulator), but such an approach has its well known inherent limitations; in particular, (basic) simulation methods hardly allow for doing sensitivity analysis, as it may already be rather time-consuming to get a reliable estimate for a single parameter instance. Therefore we have chosen to set up a model that allows for an analytical approach, or, when its evaluation turns out to be too complicated, a hybrid approach in which the role played by simulation is minimized.

As motivated by the above, TCP's widespread use and complex behavior have triggered the search for simple and transparent, yet accurate, mathematical techniques for performance analysis. Many performance models have been proposed, which can be roughly divided into *packet-level models* and *flow-level models*.

Packet-level models describe the detailed dynamics of TCP, related to the evolution of the flows' window sizes and transmission rates. Evidently, the detailed behavior of a link fed by various classes of TCP flows, is intrinsically difficult to capture, particularly due to the complex interactions between the source behavior and the congestion level. However, by assuming a constant number of greedy (i.e., persistent) flows, and after imposing some additional simplifications, explicit expressions were obtained for the flows' throughput, as a function of the packet loss probability; early references are Kelly [15], Mathis *et al.* [21], and Padhye *et al.* [23]. Noticing that in return the packet loss probability of a bottleneck link is a function of the offered load (and hence essentially also of the throughput), we obtain two equations in two unknowns, and as a result the throughput can be found, see e.g. [3, 8, 17]. Some papers explicitly model TCP's window dynamics, and allow at the same time user heterogeneity, e.g., [11, 22], but this approach is severely limited by scalability constraints.

Where packet-level models consider a constant set of persistent flows feeding into a link, flow-level models explicitly focus on the dynamics of the number of flows present. In other words: flow-level models focus on a somewhat less detailed time-scale than packet-level models, namely the time-scale at which flows arrive and depart, see for instance [20]. It is assumed that, at the moment such a flow-level transition takes place, the allocation of the transmission rates to the individual flows adapts instantly. This enables the use of *processor sharing* (PS) queues, addressed in great generality by Cohen [9]. Over the past years this class of models has gained ground as a generic description of TCP's flow-level dynamics, as advocated in, e.g., [5, 16].

It is clear that TCP's packet-level and flow-level have a strong mutual dependency, which motivates the attempts to develop a unified approach. In this respect we mention the pioneering work of Gibbens *et al.* [12], who consider the network extension of the single-link packet-level models mentioned above, enabling them to compute the throughputs for any given number of flows simultaneously present at the various routes through the network. Then they weigh these throughputs with an *a priori* supposed distribution for this number of 'concurrent flows' (Poisson, geometric), in order to 'emulate' the flow level. Though reasonable, a rigorous justification of assuming these specific flow-level distributions was lacking. This motivated why Lassila *et al.* [19] have considered ways to *derive* (i.e., to 'endogeneously determine') the flow-level distribution from the model, rather than to exogeneously impose a distribution. It is noted that [19] has succeeded in doing so in a single-link setting with homogeneous input (i.e., flows have the same job-size distribution, round-trip times, and access rates).

In the present paper we build on the results of [12, 19], but we add a number of substantial enhancements. Relative to [19], a first contribution of our work is that we allow for heterogeneous input at the flow level: several classes are distinguished (characterized by flow arrival rate, flow size, round-trip time, and access rate). Then the procedure is that we first apply packet-level models to compute the (per-class) throughputs for a fixed number of flows present, which we use as the input for a PS-type of flow-level model. It is noted that the PS model that arises in this setting has the flavor of a so-called *discriminatory*

*processor sharing* (DPS) system, which is in general notoriously hard to study analytically, see, e.g., [10]. A second improvement over [19] is that our multiple time-scale framework lends itself to being extended to network settings. The model can be used to assess the effect of user heterogeneity in often highly complex multi-link situations. Compared to [12] the major improvement is that we, as in [19], *derive* the fbw-level distribution (i.e., the joint distribution of the number of ‘concurrent fbws’ of the various types), rather than that we impose *a priori* some distribution.

It is noted that, particularly in the situation that the number of user classes grows large, it may become extremely time-consuming to numerically solve the fbw level. For those situations we propose a ‘hybrid’ method: the packet-level is solved numerically, and the resulting throughputs are used as input in the *simulation* of the fbw level. It is noted that such an approach is still substantially faster than detailed time-scale (NS2) simulations, as only the fbw-level jumps need to be simulated rather than the full packet-level dynamics. We remark that a somewhat similar hybrid approach was developed in [4]; there the simulated jumps correspond to congestion epochs.

The remainder of this paper is organized as follows. In Section 2 we introduce our packet/fbw level model for heterogeneous TCP-controlled traffic transmitted over a buffered link. Quantitative validation of the model, as reported in Section 3, is done by using the simulator NS2 [13] as a benchmark. Also a number of other numerical experiments are performed, with a strong focus on the impact of the heterogeneity in round-trip times on the throughputs experienced by the various user classes. We give some further comments on the relation with discriminatory processor sharing. Section 4 describes the extension of the single-link model of Section 2 to non-cyclic multiple-link network scenarios. Importantly, it is shown that ‘packet-level parameters’ such as buffer sizes, do have a significant impact on the way TCP allocates bandwidth. Section 5 concludes the paper.

## 2 Integrated packet/flow level modelling approach

In this section we introduce a mathematical model that describes a buffered network link fed by a fluctuating set of heterogeneous TCP fbws. As argued in the introduction, a model that captures all the details is far too complex to analyze.

A commonly used resort is to rely on *time-scale decomposition*: distinguish between multiple time-scales, solve these separately, and integrate them into the performance measures of interest. We here follow such a decomposition approach, by decoupling the *packet level* and the *flow level*. Recall that the packet level describes the performance when the link is used by a fixed set of persistent fbws, whereas the fbw level describes the fluctuations of the number of fbws simultaneously present. Interestingly, the decomposition allows us to analyze the effect of typical packet-level parameters (buffer size, round-trip times, etc.) on fbw-level performance (the average number of fbws in the system, fbw transfer times, etc.). It is stressed that our specific focus is on assessing the impact of user heterogeneities on the performance. To this end, we will introduce  $m$  classes of fbws.

**Modelling assumptions.** Each class  $i$ , for  $i = 1, \dots, m$ , is characterized by four parameters. (i) The rate at which fbws of class  $i$  are initiated is denoted by  $\lambda_i$ ; it is assumed that these arrivals follow a Poisson process. (ii) Flows of class  $i$  have an exponentially

distributed size with mean  $\mu_i^{-1}$  (packets); it is assumed throughout this paper that packets are equally sized. (iii) The ‘physical’ round-trip time of a class- $i$  packet, i.e., due to propagation and all other non-congestion-dependent factors, is given by  $\text{RTT}_i^0$  time units (in other words: the queuing delay in the buffer is *not* incorporated in  $\text{RTT}_i^0$ ). (iv) The access rate of a class- $i$  user is given by  $R_i$  (packets per time unit). Notice that the maximum window size  $W_{\max,i}$  and the round-trip time also put a limit on the users’ transmission rate:  $R_i \leq W_{\max,i}/\text{RTT}_i^0$ .

The link, to be interpreted as a *bottleneck* link, is characterized by its service speed (or link rate)  $C$  (expressed in packets per time unit) and buffer size  $B$  (packets). The service discipline is *first-in-first-out*. The  $m$  traffic classes share these common network resources. For reasons of stability, the incoming load is limited to the system’s capacity:  $\sum_{i=1}^m \lambda_i/\mu_i < C$ . Now we subsequently describe the packet level and fbw level, and describe how these allow the computation of the performance measures of our interest.

**Packet level.** In the packet-level model there is a *fixed* number  $n_i$  of *persistent* TCP fbws of class  $i$ , for  $i = 1, \dots, m$ . The main objective of the packet-level analysis is to compute, for a given vector  $\bar{n} := (n_1, \dots, n_m)$ , the throughputs  $t_1, \dots, t_m$  of the various classes.

In our approach we rely on a relation between the mean throughput, the packet-loss probability, and the round-trip time, that was derived for the case of just one class of TCP connections, see e.g. [15, 21, 23]; in our study we will rely on the formula given in [15]. Suppose there are  $n$  fbws with round-trip time  $\text{RTT}$  and access rate  $R$ , who experience a packet-loss probability  $p$ . Then the (total) throughput in the congestion-avoidance phase is approximated by

$$t(n) = \min \left\{ nR, \frac{n}{\text{RTT}} \sqrt{\frac{2(1-p)}{p}} \right\};$$

evidently, the left-hand argument of the min function simply limits throughputs to the access rate  $R$ . By borrowing the above formula, we obtain for our case of  $m$  heterogeneous traffic classes:

$$t_i(\bar{n}) = \min \left\{ n_i R_i, \frac{n_i}{\text{RTT}_i} \sqrt{\frac{2(1-p_i)}{p_i}} \right\}, \quad i = 1, \dots, m. \quad (1)$$

It is clear that the round-trip time  $\text{RTT}_i$  consists of its fixed component  $\text{RTT}_i^0$ , increased by the queuing delay experienced by class  $i$ . With  $\delta_i$  denoting the mean queuing delay of class  $i$ , we could write  $\text{RTT}_i = \text{RTT}_i^0 + \delta_i$ . Regarding loss and delay, it seems reasonable to assume that there will not be too much discrepancy between the classes, which allows us to write  $\delta_i = \delta$  and  $p_i = p$ , for  $i = 1, \dots, m$ .

Having expressed the throughput in terms of the packet loss and delay, we have to find loss and delay as a function of the throughput; clearly, having these relations at our disposal, we are able to compute the  $t_i(\bar{n})$ . To this end, we make the approximation that, at the packet level, packets (of equal size) arrive at the buffer according to a Poisson process. Hence, we can approximate the packet-loss probability and the mean delay by those of the corresponding M/D/1/ $B$  queue. In self-evident notation:

$$p \equiv p(\bar{n}) = p_{\text{M/D/1}} \left( \sum_{i=1}^m t_i(\bar{n}), C, B \right); \quad \delta \equiv \delta(\bar{n}) = \delta_{\text{M/D/1}} \left( \sum_{i=1}^m t_i(\bar{n}), C, B \right). \quad (2)$$

Techniques for computing (or approximating) both the loss probability and mean delay in M/D/1 queues are described in, e.g., [24, Section 15.1]; the mean delay is given by their Eq. (15.1.2), whereas the loss probability could be accurately approximated by exponential expansion in the spirit of their Eq. (15.1.6).

The assumption of Poisson arrivals at the packet level has been very common in the literature, see e.g. [3, 12, 17]. Given the fact that many fbws are multiplexed, and in the absence of a detailed description of the packet arrival process, we have chosen to do so. It can obviously not be justified that this choice yields the best match with the actual queueing behavior. We emphasize, however, that the methodology presented here does not critically rely on this Poisson assumption; if there is a reason to assume that some other arrival process provides a better match, one could employ the corresponding queueing model instead.

It is clear that inserting (2) into (1) yields a fixed-point problem of  $m$  non-linear equations with  $m$  unknowns (i.e.,  $t_1, \dots, t_m$ ; we suppress the  $\bar{n}$  for convenience). Now consider the right-hand side of (1) as a function of  $t_i$ , for fixed  $t_j$ ,  $j \neq i$ . Evidently, both the mean queueing delay  $\delta$  and the loss probability  $p$  increase in  $t_i$ . It can be verified easily that this implies that the right-hand side of (1) decreases in  $t_i$ , and has limit 0. As the left-hand side (i.e., the identity function) increases from 0 to  $\infty$ , we conclude that there is, for given  $t_j$ ,  $j \neq i$ , a fixed point.

**Flow level.** The fbw level describes the dynamics related to arrivals and departures of fbws (i.e, TCP connections). When there are  $\bar{n}$  fbws in the system, we assume that class  $i$  is served at a rate  $s_i(\bar{n}) := t_i(\bar{n})(1 - p(\bar{n}))$ , where  $t_i(\bar{n})$  and  $p(\bar{n})$  follow from the packet-level fixed-point equations;  $s_i(\bar{n})$  is often referred to as the class- $i$  ‘goodput’. All individual fbws of type  $i$  are assumed to get a ‘fair share’  $s_i(\bar{n})/n_i$  of the class- $i$  goodput. It is remarked that our time-decomposition entails that we implicitly assume that, when the number of active fbws changes, service rate adaptation takes place instantly. Put differently, our approach neglects the rate fluctuations at the packet level (which are due to the window-size dynamics, in response to packet losses).

As the fbw sizes were assumed to be exponential, we can model the fbw-level dynamics by a (continuous-time) Markov chain. The fact that any fbw gets a fair share of the per-class goodput, gives our model the flavor of a processor-sharing system. The single-class system can be solved explicitly; see [19], relying on the results of [9]. The user heterogeneity, as present in our model, however, makes the analysis considerably more difficult. As mentioned in the introduction, the fbw-level model that arises in this setting has the flavor of a DPS system, a class of models that is in general hard to study analytically.

We now give the transition rates of the continuous-time Markov chain governing the fbw level, with state space  $\mathbb{N}_0^m$ . The transition rate corresponding to an arrival of a type- $i$  fbw, denoted by  $q(\bar{n} + e_i | \bar{n})$ , is clearly equal to  $\lambda_i$ ; here  $e_i := (0, \dots, 0, 1, 0, \dots, 0)$ , where the 1 is put on the  $i$ th position. It can be verified easily that the transition rate from  $\bar{n}$  to  $\bar{n} - e_i$ , i.e.,  $q(\bar{n} - e_i | \bar{n})$ , equals  $\mu_i s_i(\bar{n})$  (under the proviso that  $n_i > 0$ ; otherwise  $\bar{n} - e_i$  clearly does not belong to the state space). Having these transition rates, we can compute the equilibrium distribution  $\pi(\bar{n})$ .

In case solving the system of balance equations leads to numerical problems (the iterative solution techniques may yield oscillations), these can be overcome by using simulation at the fbw level. This procedure could be characterized as a *hybrid* approach, between computation and simulation. An alternative could be to simulate *both* packet-level and

fbw-level dynamics, for instance by using NS2, but such an approach is substantially slower. Clearly, the main difference lies in the fact that our integrated approach does not need to simulate the detailed packet-level dynamics; these are summarized by the (computed) throughput values. In this respect there is an interesting similarity with the hybrid simulation/computation approach proposed in [4]; an important difference is that in our approach the fbw-level fluctuations (fbw arrivals, fbw departures) are the jump epochs, whereas in [4] this role is played by the congestion epochs.

### 3 Validation; impact of user heterogeneity

This section presents the quantitative validation of our packet/fbw level approach by using NS2 simulations; we have chosen to use NS2 [13] as a benchmark, motivated by the fact that it mimicks TCP at a detailed time-scale. Hence, comparing results obtained by using our methodology with results obtained under NS2, we assess to what extent it is justified to replace the packet-level dynamics by a static bandwidth allocation according to the rates  $s_i(\bar{n})$  (when there are  $\bar{n}$  fbws in the system), as is done in our approach.

We also perform a number of further experiments assessing the impact of heterogeneity on the performance bias (i.e., the difference between the classes in, e.g., the mean file transfer delay) with emphasis on the impact of packet-level parameters (such as the buffer size).

**Scenarios.** We consider the single bottleneck scenario as in Section 2, with  $m = 2$  traffic classes. Recall that the number of concurrently active fbws within each class ( $n_1$  and  $n_2$ ) is varying over time due to fbw arrivals and departures.

In our experiments we chose the loads per class 1 and 2 to be equal:  $\lambda_1/\mu_1 = \lambda_2/\mu_2 = \rho/2$ , where  $\rho < C$  is the system load. We concentrate on the heterogeneity with respect to round-trip times and file sizes; we assume that all connections have the same access link rate, i.e.,  $R_1 = R_2$ . We set the link rate  $C$  equal to 10 [Mbit/s]. All transmitted packets have a size of 1500 [bytes] (cf. MTU size).

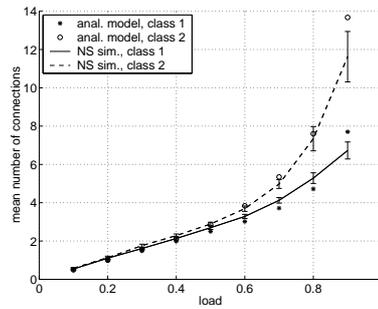
In practical situations round-trip times vary, roughly speaking, over a range of 50 to 200 msec. Therefore we have chosen to consider round-trip times of 50, 100, and 200 msec. We study the influence of small and large buffer sizes, i.e., 10 and 50 [packets], and access rates, i.e., 1 and 2 Mbit/sec. In the sequel we provide a comparison of results for a full permutation of all the above parameters, where, in addition, both classes have different round-trip times. We have set the mean file size to 500 [packets] for all fbws. Recall that we took a constant packet size of 1500 bytes.

**Numerical results.** For different values of the system load  $\rho$ , Figure 1 shows the mean number of active connections per class  $n_i^*$ ; the graph contains both the values obtained by using our method, and the values obtained by using NS2 (including their confidence intervals).

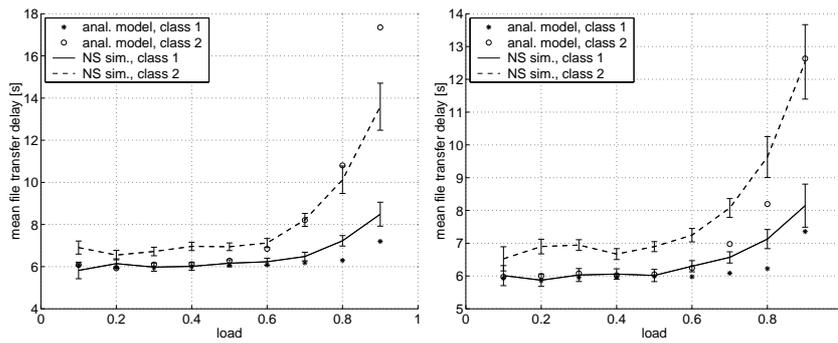
In Figures 2 and 3 we have focused on the perhaps somewhat more appealing performance measure of the mean fbw transfer delay  $D_i$ , related to  $n_i^*$  through Little's law  $D_i = n_i^*/\lambda_i$ . Bearing in mind the complex TCP dynamics, the figures show that our model results coincide remarkably well with the NS simulation results, and it does so for a wide set of parameter combinations.

We now discuss in detail the accuracy of our packet/fbw level modelling approach. We mention the following observations:

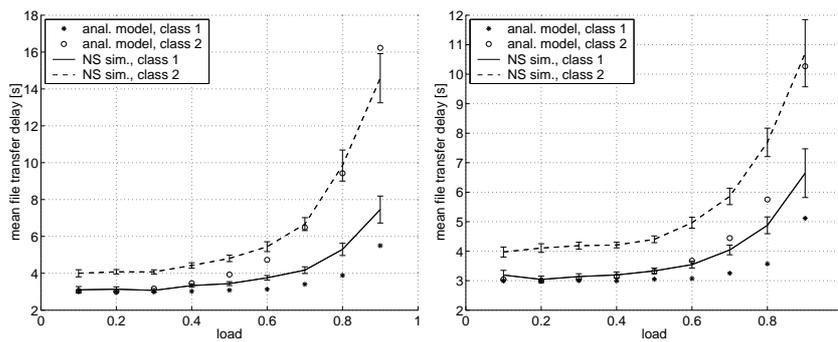
- For low values of the load, our model predicts nearly identical performance for class 1 and class 2. The NS2 simulations, however, show that these results are systematically too optimistic and that the flows with the short round-trip time experience the smaller delay. We suspect that this behavior is caused by the fact that our method does not incorporate the effects of TCP's slow-start phase: the time-scale decomposition entails that we do as if the flow is, after being initiated, immediately in its 'stationary behavior'. It is noted



**Fig. 1.** Mean number of connections for two classes, under varying load. Access rates 1 [Mbit/sec.],  $RTT_1^0 = 100$  and  $RTT_2^0 = 200$  [msec.], buffer size 10 [packets].



**Fig. 2.** Mean file transfer delay [sec.]; access rates 1 [Mbit/sec.],  $RTT_1^0 = 50$  and  $RTT_2^0 = 200$  [msec.], buffer size left panel: 10 [packets], buffer size right panel: 50 [packets].



**Fig. 3.** Mean file transfer delay [sec.]; access rates 2 [Mbit/sec.],  $RTT_1^0 = 50$  and  $RTT_2^0 = 200$  [msec.], buffer size left panel: 10 [packets], buffer size right panel: 50 [packets].

that, during the slow-start phase, the round-trip times play a crucial role, as they dictate how fast the window size can grow. In [19] the effects of the slow-start phase were successfully compensated, in order to avoid too optimistic estimates; one could pursue the development of such a procedure for the model under study.

- When comparing Fig. 2 (access rate of 1 Mbit/sec.) with Fig. 3 (access rate of 2 Mbit/sec.), it could be concluded that for low values of the load the transfer delays are not so much determined by the round-trip times, but rather by the access rate constraints. This is reflected by the fact that the transfer delays in Fig. 2 are (roughly) two times as high as in Fig. 3.

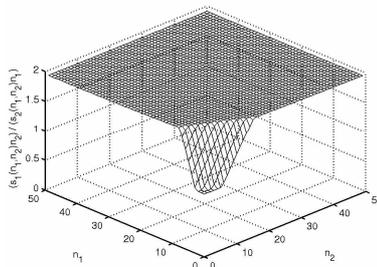
- The numerical results (also the ones not shown here) indicate that our model (particularly for class 2) is less accurate for large buffers ( $B = 50$ ) than for small buffers ( $B = 10$ ). It is expected that this is due to the fact that the real packet arrival process is often substantially burstier than Poisson, leading to errors in the approximation of the loss probability  $p$  and queuing delay  $\delta$ , which are typically smaller for a small buffer than for a large buffer.

Refinements of our approach, which take into account the issues identified above, are subjects for future research.

**Quantification of the bias; equalizing effects.** We finally say some words on the quantification of the bias, in the situation of heterogeneous round-trip times. There are several papers on this issue, see for instance [1, 18]. We here study the claim that per-fbw throughputs are inversely proportional to the fbw's round-trip time, in line with formula (1), see, e.g., [15, 21, 23]. In case of two classes sharing a bottleneck link, this would mean that

$$\frac{s_1(\bar{n})/n_1}{s_2(\bar{n})/n_2} \approx \frac{RTT_2}{RTT_1}. \quad (3)$$

Consider a situation with  $RTT_2^0/RTT_1^0 = 2$ . We plot the left hand side of (3), by using the numbers obtained in our packet-level model, see Figure 4.



**Fig. 4.** ‘Normalized goodput ratio’  $(s_1(n_1, n_2)/n_1)/(s_2(n_1, n_2)/n_2)$  plotted as function of the states  $(n_1, n_2)$ ; access rates 1 [Mbit/sec.],  $RTT_1^0 = 100$  and  $RTT_2^0 = 200$  [msec.], buffer size 10 [packets].

Indeed, the ratio is nearly constant for somewhat larger values of  $n_1, n_2$ , and has indeed value 2; for ‘low’ states the ratio is significantly smaller. In fact, close to the origin the ratio is nearly 1, as such a small number of connections (with limited access rate) is not able to ‘fill up the link’.

There are circumstances under which the performance asymmetries tend to be ‘equalized’. In the first place, this is clearly the case for low and medium loads, and limited access rates. For such loads the most probable states are relatively close to the origin, and Figure 4 then indicates that the ‘normalized goodput ratio’ will be close to 1.

A second factor that has impact on this ‘equalizing effect’ is the buffer size. Table 1 shows the limiting value of the ‘normalized goodput ratio’, i.e.,  $(s_1(\bar{n})/n_1)/(s_2(\bar{n})/n_2)$  for large  $n_1, n_2$ . We do so for varying (i) ratio of the (‘physical’) round-trip times  $r := \text{RTT}_2^0/\text{RTT}_1^0$ , and (ii) buffer size. The link capacity is still 10 [Mbit/sec.].

**Table 1.** ‘Normalized goodput ratio’, buffer sizes are in packets.

	$B = 10$	$B = 50$
$r = 2$	1.90	1.63
$r = 4$	3.45	2.38

From the table we conclude that for small buffers the ‘normalized goodput ratio’ is close to the ratio of the round-trip times, whereas for larger buffers this match is less good. An explanation lies in the fact that for larger buffers the round-trip times are increasingly determined by the queueing delay (rather than the ‘physical’ delay), and this queueing delay is the same for both classes.

We conclude that it is not always accurate to assume goodputs inversely proportional to the round-trip time. It also entails that it could be inaccurate to approximate the model with heterogeneous users with discriminatory processor sharing [10], with weights inversely proportional to the classes’ round-trip times, as done in, e.g., [2]. Our model nicely captures the ‘equalizing effect’ of the access rate limitation and the buffer size on the throughputs, as experienced by fbws with different (‘physical’) round-trip times; it is noted that similar properties were observed in the simulations performed in [5].

## 4 Extension to networks

In this section we make a first step towards extending our single-link framework to a network setting. Model and analysis are presented in Section 4.1. The approach borrows elements of [12], but we remark again that the crucial difference between our approach and [12] is, that our model finds the distribution of the network population ‘endogeneously’. In Section 4.2, we consider a few examples, and investigate how these compare to other results on rate allocation.

**Modelling and analysis.** Consider a non-cyclic network, consisting of  $k$  links, characterized by their link rates  $C_j$  and buffer sizes  $B_j$ . Flows of class  $i$  subsequently pass a number of links. We say that  $j \in \mathcal{R}_i$  if class  $i$  uses link  $j$ . Also, we say that  $j_1 \prec_i j_2$  if  $j_1, j_2 \in \mathcal{R}_i$ , and in addition  $j_1$  lies on this route *before*  $j_2$ .

The approach relies again on decoupling the packet level and the fbw level. We first determine the throughput of class  $i$ , for a given user population  $\bar{n}$ ; then this serves as input for the continuous-time Markov chain model that describes the fbw level. The generalization of the throughput formula (1) is, for  $i = 1, \dots, m$ ,

$$t_i(\bar{n}) = \min \left\{ n_i R_i, \frac{n_i}{\text{RTT}_i^0 + \sum_{j \in \mathcal{R}_i} \delta_j} \sqrt{\frac{2 \left(1 - \sum_{j \in \mathcal{R}_i} p_j\right)}{\sum_{j \in \mathcal{R}_i} p_j}} \right\}. \quad (4)$$

Here  $p_j$  is the loss probability at link  $j$ , and  $\delta_j$  the mean delay at link  $j$ ; we assume that the per link loss probability and mean delay is constant across the user classes.

In return, the loss probabilities and mean delays depend on the load offered to the links. We can write

$$p_j \equiv p_j(\bar{n}) = p_{M/D/1} \left( \sum_{i:j \in \mathcal{R}_i} t_i(\bar{n}) \cdot \left( 1 - \sum_{k \prec_i j} p_k(\bar{n}) \right), C_j, B_j \right); \quad (5)$$

$$\delta_j \equiv \delta_j(\bar{n}) = \delta_{M/D/1} \left( \sum_{i:j \in \mathcal{R}_i} t_i(\bar{n}) \cdot \left( 1 - \sum_{k \prec_i j} p_k(\bar{n}) \right), C_j, B_j \right). \quad (6)$$

Notice that the load imposed on link  $j$  is ‘thinned’, along the various routes  $i$ , by the losses in its predecing links (i.e.,  $k$  such that  $k \prec_i j$ ); the formula uses, as in [12], the approximation that losses at the various links are independent, cf. also [14], and the well known approximation  $\prod_i (1 - x_i) \approx 1 - \sum_i x_i$  for small  $x_i$ . Combining (5) and (6) with the throughput formula (4) again yields  $m$  equations in  $m$  unknowns, which can be solved numerically.

The fbw level can again be done by constructing a continuous-time Markov chain; it is obvious that again it can be solved by either solving the balance equations, or by simulating an  $m$ -dimensional random process. It is noted that the fbw-level procedure remains essentially the same, and that the network topology is just reflected by the packet-level computations.

**Examples; rate allocation in TCP.** It has been claimed that the transmission rates allocated in TCP’s congestion avoidance are well approximated by the optimizing  $t_i$ ,  $i = 1, \dots, m$  in

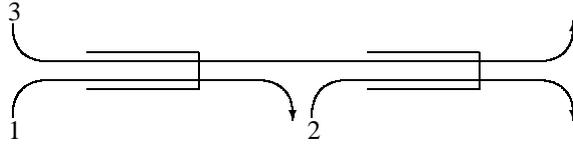
$$\min_{t_1, \dots, t_m} \sum_{i=1}^m \left( \frac{1}{\text{RTT}_0^i} \right)^2 \frac{n_i}{t_i} \quad \text{under} \quad \sum_{i:j \in \mathcal{R}_i} t_i \leq C_j;$$

see for instance [6, Eq. (3)]. For a single link fed by  $m$  heterogeneous classes, when performing the optimization, one obtains, with  $\kappa_i$  defined as  $(\text{RTT}_0^i)^{-1}$ , that  $t_i(\bar{n})$  equals  $C n_i \kappa_i / \sum_{j=1}^m n_j \kappa_j$ , i.e., the rate allocation is according to DPS with weights inversely proportional to the round-trip times. We have seen in the previous section that this approximation is particularly accurate when buffers are relatively small.

Now consider the somewhat more involved case of a network with multiple links. For ease we concentrate on the simplest, non-trivial model, as depicted in Figure 5, since this already shows a number of features that are not present in the single-link case. Let there be two links, both of rate  $C$ ; let type 1 use link 1, type 2 use link 2, and type 3 use both links. It can be verified that the above minimization now yields that  $t_1(\bar{n}) = t_2(\bar{n})$  and

$$t_1(\bar{n}) = \frac{C \cdot \sqrt{(n_1 \kappa_1)^2 + (n_2 \kappa_2)^2}}{n_3 \kappa_3 + \sqrt{(n_1 \kappa_1)^2 + (n_2 \kappa_2)^2}}; \quad t_3(\bar{n}) = \frac{C \cdot n_3 \kappa_3}{n_3 \kappa_3 + \sqrt{(n_1 \kappa_1)^2 + (n_2 \kappa_2)^2}}. \quad (7)$$

We see that this allocation rule is not quite DPS, and we wonder whether it coincides with the throughputs realized in our model and in NS2. Interestingly, even when  $\text{RTT}_1^0 \neq \text{RTT}_2^0$ , the above allocation indicates that the fbws of type 1 enjoy the same throughput as the fbws of type 2. In practice, however, one expects that this heterogeneity of round-trip times should have impact.



**Fig. 5.** Topology of two-link example.

To study these effects we have performed two experiments. In the first we vary the heterogeneity between the round-trip times. This is done by choosing  $RTT_1^0 = 1$  and  $RTT_3^0 = 3$ , and  $RTT_2^0 \in \{0.5, 1, 2, 5, 10\}$ . We have fixed the numbers of users  $n_1 = n_2 = n_3 = 10$ , the buffer size  $B_1 = B_2 = 10$  packets, and the link rates  $C_1 = C_2 = 100$ . For ease we assume that no access rate limitations are imposed. The throughputs as derived by our model, as well as those based on (7), are tabulated in Table 2.A. We conclude that the heterogeneity does have a significant impact on throughputs; in particular, there could be a substantial difference between  $t_1$  and  $t_2$ , which was not predicted by the rate allocation models of, e.g., [6]. Interestingly, class 3 considerably benefits when  $RTT_2^0$  increases.

The second experiment studies the impact of the buffer size on the bias. We have varied  $B = B_1 = B_2$ , and fixed  $RTT_0^0 = 5$ ; the other parameters are the same as in the first experiment. Table 2.B shows the same ‘equalizing effect’ as we have seen before: for small buffers the bias is very pronounced, whereas for larger buffers, class 1 and class 2 see essentially the same round-trip delay (mainly queueing delay), and as a consequence the bias disappears.

**Table 2.** A. (left) Throughputs  $t_i(\bar{n})$  as a function of  $RTT_2^0$ ; the allocation according to (7) is given between parentheses. B. (right) Throughputs  $t_i(\bar{n})$  as a function of  $B$ ; the allocation according to (7) would be  $t_1 = 75.4$ ,  $t_2 = 75.4$ , and  $t_3 = 24.6$ .

$RTT_2^0$	$t_1$	$t_2$	$t_3$	$B$	$t_1$	$t_2$	$t_3$
0.5	72.4 (87.0)	77.1 (87.0)	12.2 (13.0)	2	43.9	22.3	13.7
1	68.3 (80.9)	68.3 (80.9)	16.7 (19.1)	5	56.9	39.9	18.5
2	65.6 (77.0)	60.7 (77.0)	19.8 (23.0)	10	64.2	52.9	21.3
5	64.3 (75.4)	53.0 (75.4)	21.3 (24.6)	20	68.4	62.0	24.6
10	64.0 (75.1)	48.1 (75.1)	21.6 (24.9)	50	69.9	67.2	26.5

## 5 Concluding remarks

We have developed a versatile TCP performance model that explicitly captures user heterogeneity. This multiple time-scale model integrates the dynamics at the packet level with those at the fbw level. It is relatively simple, and allows for straightforward numerical evaluation. The accuracy of the model was validated by using TCP simulator NS2. In extensive numerical experiments, we have studied the impact of heterogeneity in the round-trip times on user-level characteristics (throughputs, fbw transmission times). Interestingly, we have seen that the asymmetry caused by this heterogeneity is somewhat mitigated if the access rates are small (and, in addition, the load is relatively low), or the buffer is large.

We have pointed out how the single-link model can be extended in a straightforward way to the general network setting. It is noted that settings with heterogeneous users (in particular networks) pose interesting questions related to rate allocation. Like in the single-link case, we have seen that the size of the buffers does affect the rates allocated by

TCP: again for large buffers the performance bias disappears. The network model opens up the possibility of a careful assessment of those phenomena; it is remarked that most existing rate allocation results, see, e.g. [6], do not take into account buffering.

## References

1. E. Altman, C. Barakat, E. Laborde, P. Brown, and D. Collange. Fairness analysis of TCP/IP, Proc. IEEE Conference on Decision and Control, pp. I: 61-66, 2000.
2. E. Altman, T. Jiménez, and D. Kofman. DPS queues with stationary ergodic service times and the performance of TCP in overload, Proc. INFOCOM 2004, 2004.
3. K. Avrachenkov, U. Ayesta, E. Altman, P. Nain, and C. Barakat. The effect of the router buffer size on the TCP performance, Proc. LONIIS International Seminar on Telecommunication Networks and Teletraffic Theory, pp. 116-121, 2002.
4. F. Baccelli and D. Hong. Flow level simulation of large IP networks, Proc. INFOCOM 2003, 2003.
5. S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié, and J.W. Roberts. Statistical bandwidth sharing: a study of congestion at flow level, Proc. ACM SIGCOMM 2001, pp. 111-122, 2001.
6. T. Bonald and L. Massoulié. Impact of fairness on Internet performance, Proc. ACM SIGMETRICS 2001, pp. 82-91, 2001.
7. P. Brown. Resource sharing of TCP connections with different round trip times, Proc. INFOCOM 2000, pp. 1734-1741, 2000.
8. T. Bu and D. Towsley. Fixed point approximation for TCP behavior in an AQM network, Proc. ACM SIGMETRICS 2001, pp. 216-225, 2001.
9. J.W. Cohen. The multitype phase service network with generalized processor sharing, Acta Informatica, Vol. 12, pp. 245-284, 1979.
10. G. Fayolle, I. Mitrani, and R. Iasnogorodski. Sharing a processor among many classes, J. ACM, Vol. 27, pp. 519-532, 1980.
11. N. van Foreest, M. Mandjes, and W. Scheinhardt. A versatile model for asymmetric TCP sources, Proc. ITC 18, pp. 631-640, 2003.
12. R. Gibbens, S. Sargood, C. van Eijl, F. Kelly, H. Azmoodeh, R. Macfadyen, N. Macfadyen. Fixed-point models for the end-to-end performance analysis of IP networks, Proc. 13th ITC Specialist Seminar, 2000.
13. Homepage, Network Simulator II, <http://www.isi.edu/nsnam/ns/>
14. F. Kelly. Blocking probabilities in large circuit-switched networks. Adv. Appl. Prob., Vol. 18, pp. 473-505, 1986.
15. F. Kelly. Mathematical modeling of the Internet, Mathematics Unlimited - 2001 and Beyond, pp. 685-702, Springer-Verlag, Berlin, 2001.
16. A. Kherani and A. Kumar. Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet. Proc. INFOCOM 2002, pp. 1014-1023, 2002.
17. P. Kuusela, P. Lassila, J. Virtamo, and P. Key. Modeling RED with idealized TCP sources. Proc. IFIP ATM & IP 2001, 2001.
18. T.V. Lakshman and U. Madhoo. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. IEEE/ACM Trans. Netw., Vol. 5, pp. 336-350, 1997.
19. P. Lassila, H. van den Berg, M. Mandjes, and R. Kooij. An integrated packet/flow model for TCP performance analysis, Proc. ITC 18, pp. 651-660, 2003.
20. L. Massoulié and J.W. Roberts. Arguments in favour of admission control for TCP flows, Proc. ITC 16, pp. 33-44, 1999.
21. M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm, Comp. Comm. Rev., Vol. 27, pp. 67-82, 1997.
22. V. Misra, W. Gong, and D. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED, Proc. ACM SIGCOMM 2000, pp. 151-160, 2000.
23. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation, Proc. ACM SIGCOMM 1998, pp. 303-314, 1998.
24. J. Roberts, U. Mocci, and J. Virtamo. *Broadband network teletraffic: Final report of action COST 242*, Springer, Berlin, 1996.