Enhancement of an integrated packet/flow model for TCP performance¹

R. de Haan*, J.L. van den Berg*, R.E. Kooij², R.D. van der Mei* and A.P. Zwart *

* Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
* TNO Telecom, Brassersplein 2, 2612 CT Delft, The Netherlands
* University of Twente P.O. Box 217, 7500 AE Enschede, The Netherlands
* Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Abstract

Processor sharing (PS) models nicely capture the bandwidth sharing and statistical multiplexing effect of TCP at the flow-level. However, these ‘rough’ models do not provide insight into the impact of packet-level parameters (round trip time, buffer size) on performance metrics such as throughput and flow transfer times. In a previous paper an analytical, integrated packet/flow model was developed, exploiting the advantages of PS approach at the flow-level, while incorporating the most significant packet-level effects. In the present paper we propose an enhancement of this model, which is validated through extensive NS simulations. The numerical results show that this enhancement leads to significantly better performance approximations than the originally proposed model.

Keywords: TCP, performance modelling, On-Off model, file download times.

Relevant technical sessions: Network and Service Management, NGN architectures, Broadband access.

1 Introduction

For millions of people all over the world the Internet takes a prominent position in everyday life. The youngest generation grows up with this new medium and can not even imagine a life without it. The possibilities of the Internet seem endless. People use the Internet as a medium for e-mail services, on-line banking, to order the latest books or CDs on-line, to download music files, to exchange pictures, etc. However, for the end-user at home the underlying structure of this still booming medium is invisible. Within this structure an important role is played by the Transmission Control Protocol (TCP), as nowadays the major part (>80%) of the data traffic over the Internet is controlled by TCP. It is also a well-known fact that the performance of TCP has a significant influence on the end-user’s perceived quality. Hence, the transmission delay, or from a user’s point of view, simply the download time of a web page or data file is a measure of particular interest.

Many studies have been investigating performance models for TCP. These models can roughly be categorized in two disjoint subsets: (i) flow-level models and, (ii) packet-level models. Both types of models aim at modelling specific TCP characteristics. Flow-level models capture the effects of variation in the number of TCP flows. These models assume that each flow gets an equal share of the total available bandwidth. This implies that changes in the number of flows are translated instantaneously into the individual assigned capacity of each flow. The analysis of the flow-level model can be done using Processor Sharing (PS) models (see, e.g. [3], [13]). A drawback, however, is that important model parameters, such as round trip time (RTT) and buffer size, are not taken into account, while these are known to have a major impact on TCP performance. On a smaller time scale the packet-level model describes more detailed events, e.g. the expansion and shrinkage of the congestion window, packet losses and buffering delays. Packet-level models typically assume a fixed number of flows and thus neglect the dynamics of arriving and departing flows. A positive aspect is that the throughput can be expressed as a function of the packet loss and the RTT (see [7], [12]). Recently, a new modeling approach based on fluid models has been introduced. Fluid models adopt an abstract deterministic description of the average network dynamics through a set of differential equations, neglecting the short-term, packet-by-packet description of the stochastic network dynamics. The resulting set of (partial) differential

¹ This work is partially funded by the Dutch Ministry of Economic Affairs under the programme ‘Technologische Samenwerking ICT Doorbraakprojecten’, project TSIT 2031 EQUANET.
² Corresponding author: e-mail: r.e.kooij@telecom.tno.nl, fax: +31 15 285 7057.
equations is the solved numerically to obtain estimates of the time-dependent network behaviour. The main benefit of fluid models resides in the fact that the number of differential equations to be solved is independent of the number of TCP flows and of link capacities, when considering scenarios comprising only long-lived TCP flows. A main drawback of the fluid modeling approach is the fact that the models do not capture the fluctuations in the number of flows, which in known to have a significant effect of TCP performance in many cases. We refer to [2], [6], [9] and [10] (and references therein) for details.

Recent work of Lassila et al. (see [8]) proposes the integration of the packet- and flow-level model in order to exploit the advantages of both models. In [8] a simple network model consisting of a single link with a finite buffer has been studied. TCP flows are assumed to arrive at the link according to a Poisson process. The approach of [8] is to compute the throughput at packet-level for a fixed number of flows and next to use these throughput values as input for the flow-level model. The flow level characteristics are modelled by a Generalized Processor Sharing (GPS) model. User throughputs, conditioned on the number of flows in the system, are obtained from the models of [7], [12] in conjunction with an M/D/1/K queueing model in order to estimate packet loss characteristics, which are used as a parameter in the model of [12]. However, in some cases this model is still not very accurate in predicting the mean download times.

In this paper we propose an enhancement for the integrated TCP model of [8]. Our contribution mainly concentrates on the improvement of the packet loss probabilities at the packet-level. In particular, instead of using an M/D/1/K queueing model to estimate the packet loss and delay figures, we deploy an On-Off model to describe the packet arrival process at the link more accurately, and hence, to obtain a better, but still analytically tractable, approximation of the packet loss probability. The mean transmission delays following from our integrated approach are verified by extensive NS (see [11]) simulations. The results show that the application of the On-Off model significantly improves the integrated model of [8] and in particular yields much better predictions for high access link rates.

This paper is organised as follows. Section 2 briefly describes the previous integrated packet/flow TCP performance model (see [8]). In Section 3 we explain our packet-level approach. The numerical results following from this approach are presented in Section 4. Finally, in Section 5 we present our conclusions and mention a number of topics for further research.

2 Modelling approach

The network model considered in [8] is a simple model consisting of a single shared link. Requests for flows (i.e. data files) arrive according to a Poisson process with rate λ and the flows are fetched from a certain server. Each flow is assumed to arrive at the link independent of any other flow in the network. The shared link is equipped with a buffer of size K (in packets) and has a transmission speed c (in Mb/s). The lengths of the data files (in packets) are assumed to be generally distributed with finite mean $\frac{1}{\mu_j}$. It is further assumed that the access link rate is limited to r and that all flows have identical RTTs.

The idea of the integrated packet/flow model proposed in [8] is first to determine the throughput at packet-level for a fixed number of flows, based on detailed model properties (e.g., RTT and buffer size) and then to use these throughput values as input for the flow-level model.

The packet-level model of [8] considers a fixed number of sources actively transmitting data over the shared link. Individual flows are assumed to have identical RTTs and are sent over access links with rate r. Various studies have shown that for TCP the throughput (rather than the goodput) and the loss probability are related by a square root formula. The throughput equation adopted in [8] is the following (see [7]):

$$t_n = \frac{n}{R T T} \sqrt{\frac{2(1-p_n)}{p_n}},$$

where n is the number of flows, $p_n$ the loss probability and $t_n$ the throughput given n active flows.

To apply (1), $p_n$ has to be determined. This is done in [8] by using a simple iterative method. Equation (1) shows the dependence of $t_n$ on $p_n$. However, the loss probability $p_n$ also depends on $t_n$. In [8] it is assumed that the packets arrive at the shared link according to a Poisson process. Next, $p_n$ can be chosen equal to the loss probability in a M/D/1/K queue with load $t_n$ and server capacity c. Furthermore, the throughput is limited by the collective access rate of all active flows, i.e. $n \cdot r$, which results in the following fixed-point equation for the aggregate throughput at packet-level:
\[ t_n = \min\{nr, \frac{n}{RTT} \left( \frac{2(1-p(t_n))}{p(t_n)} \right) \}, \]

where we use the notation \( p(t_n) \) to express the dependence of \( p_n \) on \( t_n \). Note that the right-hand side is monotonously decreasing in \( t_n \), which guarantees the existence of a unique fixed-point of this equation.

The throughput \( t_n \) and the loss probability \( p_n \) following from this equation result in an aggregate goodput \( s_n \) at packet-level, where \( s_n = t_n(1-p_n) \). Subsequently, at the flow-level the GPS model with the state-dependent service rate set equal to \( s_n \) (when \( n \) flows are present), is solved. The steady-state probabilities of the GPS model can explicitly be obtained (see [4]), yielding the average number of active flows \( E[N] \). Finally, the mean transmission delay \( E[D] \) follows by applying Little’s law:

\[ E[D] = \frac{1}{\lambda} E[N], \]

where \( \lambda \) is the arrival rate of the flows at the shared link.

3 Application of the On-Off model

3.1 Motivation

In the modelling approach discussed in Section 2, the loss probability was estimated on the basis of the M/D/1/K model. The motivation for this choice was that the aggregated packet arrival process may be well approximated by a Poisson process. A limiting factor of this model, however, is that it does not take into account the correlation structure in the packet arrivals within individual TCP flows. It is expected that this correlation between the individual packet arrival instances, plays an important role in the analysis of the TCP traffic. To obtain a model that gives a better representation of the real traffic streams and moreover includes the most relevant model parameters, we have taken a closer look at the real TCP traffic pattern by means of simulation. An examination of the traffic shows that a kind of pattern in the transmission instants arises due to restrictions on the window size. These restrictions are caused by, e.g. settings for the maximum advertised window or by loss events. As a consequence, a source cannot continuously transmit data and in this way periods of data transmission will be alternated by silent periods (i.e. periods in which no data is transmitted). The translation of these observations to a mathematical model that actually contains these traffic properties yields an On-Off model. The On-Off traffic model seems very appropriate, since it assumes an alternating data transfer too. Hence, a queueing model with aggregated On-Off input traffic streams will be a better representation of the TCP behaviour at packet-level than the M/D/1/K queue.

3.2 Model description

We consider the On-Off model, sometimes indicated as fluid flow model or fluid queue, as described by Anick et al. [1], except that we apply a finite buffer variant here. The On-Off model consists of a number of On-Off sources that share a common link. This shared link is provided with a finite buffer to control the incoming traffic. Each source can be in two states, On or Off; each state corresponds to a certain activity: either the source is sending data (i.e. On-period) or the source is not sending any data at all (i.e. Off-period). We assume that the length of the Off-period as well as the length of the On-period are exponentially distributed. The transmission rate of each source, the so-called peak rate, is assumed to be constant during an On-period and equal for all sources. Furthermore, we suppose that the On- and Off-periods of all sources are identically distributed and that the sources are mutually independent. The data sent by the individual sources is routed to a shared link with a fixed link capacity; henceforth, we indicate this link as bottleneck link. Depending on the model parameters, congestion can arise at the bottleneck link and data can get lost or delayed. The loss and delay values are of specific interest to estimate the throughput at the packet-level, because these values are used as input for the fixed-point equation (2).

3.3 Setting the model parameters

The goodput outcomes \( s_n \), computed by the packet-level model, are the connection between the packet- and the flow-level. In fact, all packet-level information is passed on to the flow-level model only through
this variable. The goodput values follow from the fixed-point equation (2). The RTT in this equation consists of a fixed part $RTT_0$ (propagation and transmission delays) and a variable part $d(t_n)$:

$$RTT = RTT_0 + d(t_n),$$  \hspace{1cm} (4)

where $d(t_n)$ is the queuing delay following from the On-Off model with $n$ sources active and offered load $t_n$. In [8] the loss probability $p_n$ and the queuing delay $d_n$ are obtained by adopting a M/D/1/K queue with load $t_n$ and link rate $c$. We propose to obtain these measures by assuming the packets to arrive at the shared link according to the On-Off model as described above.

In order to obtain the relevant measures $(p_n, d_n)$ from the On-Off model we follow the approach of Tucker [14]. Tucker’s model contains the following parameters: $\theta$ and $\nu$, the parameters of the exponential distribution for the On- and Off-period respectively, $m$ the size of the buffer, $N$ the number of input sources and $C$ the shared link capacity. The peak rate of the sources is assumed to be equal to 1.

The assignment of values to the parameters $m$, $N$ and $C$ is trivial. We take $m$ equal to $K$ and $C$ to the multiplexing ratio (bottleneck link rate vs. access link rate, i.e. $c/r$). $N$ is taken equal to $n$, the fixed number of flows for which we compute the goodput at packet-level. It remains to find appropriate values for $\theta$ and $\nu$.

These values are determined by taking a closer look at the fixed-point equation. The aggregate throughput $t_n$ implies that on average for each individual source the throughput equals:

$$t_{n, individual} = \frac{t_n \cdot C}{n}. \hspace{1cm} (5)$$

Besides, for the traffic load of an individual flow we have:

$$t_{n, individual} = \frac{E[On]}{E[On] + E[Off]} = \frac{\theta_n}{\theta_n + \nu_n}, \hspace{1cm} (6)$$

where $E[On]$ and $E[Off]$ are the mean duration of an On- and Off-period, respectively, and the subscript $n$ shows the dependence of $\theta$ and $\nu$ on $n$. Now $\theta_n$ can be expressed as a function of $n$, $t_n$, $\nu_n$ and $C$:

$$\theta_n = \frac{\nu_n \cdot t_{n, individual}}{1 - t_{n, individual}} = \frac{\nu_n \cdot t_n \cdot C}{n - t_n \cdot C}. \hspace{1cm} (7)$$

This leaves us to set a value for $\nu_n$, which determines the mean duration of the On-period. Notice that the alternating On- and Off-periods arise because a source is waiting for an acknowledgement, while it is already sending its maximum number of packets. From this observation we can derive a good estimation for the mean duration of an On-period. As the maximum number of packets that are sent and not yet acknowledged equals the window size $w$, it seems appropriate to choose $\nu_n$ inversely proportional to $w$, the mean window size. Further notice that the duration of the On-period also depends on the access rate $r$, thus $\nu_n$ is defined as follows:

$$\nu_n := \frac{r}{w}. \hspace{1cm} (8)$$

Note that this implies that for a higher access rate $r$ the mean On-period will shorten.

Next, it is left to obtain an appropriate estimation for the mean window size $w$. Assume that all the packets in a window are sent within the round trip time. If we then study the right-hand side of (2) more closely, we can redefine $t_n$ as the number of packets sent each round trip time divided by the duration of the round trip time. For an individual source the numerator yields the expression $\sqrt{2(1 - p)/p}$ and henceforth we will interpret this square root as the mean window size $w$, thus:

$$w = \sqrt{\frac{2(1 - p)}{p}}. \hspace{1cm} (9)$$

In conclusion notice that we implicitly assume that all packets within a window are sent back-to-back as one flow of packets.

Finally, we use the results in Tucker [14] to obtain the required values for the loss probability and the queuing delay given the appropriate input parameters for the On-Off model. With this we can find the fixed-point for the goodput at packet-level by using (2). Note that the uniqueness and existence of this point is still guaranteed, since both $p_n$ and $d_n$ are increasing in $t_n$. 
4 Numerical results

In this section we outline the results of the On-Off approach for the integrated model; henceforth, this model is referred to as TCP-OnOff model. We will refer to the integrated model of [8] as TCP-M model. The model outcomes are verified by extensive NS simulations. We have used identical simulation scripts as in [8].

The simulation results are obtained by performing 100 independent simulation runs. In each run, which starts and ends with an empty system, 200 file requests are simulated. To pursue the investigation of steady-state behaviour at the bottleneck link, a certain warm-up and cool-down period are taken into account. This comes down to neglecting the transmission delay outcomes for the first and the last 20 files. Hence, the delay of each run is based on the delay outcomes for 160 files. Finally, the mean download time follows by taking the average delay of the 100 individual runs.

The results presented in [8] have demonstrated that the packet/flow level model presented there is particularly accurate in scenarios with small round trip times (e.g., in fixed core networks), but is less accurate for models with relatively large round trip times (e.g., in mobile networks). For this reason, the enhancements presented in the present paper are particularly useful for networks with relatively large round trip times. Consequently, the numerical investigation of the accuracy of the approximations presented below is focused on networks in which the round trip times are significant.

In our experiments we have used the following parameter settings: bottleneck link capacity $c = 10$ Mb/s, mean file size $\frac{1}{\rho_f} = 1000$ packets, packet size $\frac{1}{\rho_p} = 1500$ bytes fixed (1460 bytes of data and 40 bytes for the header). The maximum advertised window was set to $W_{max} = 1000$ packets to ensure that the maximum throughput is not determined by the maximum window size (see also the remark at the end of this section).

In the conducted experiments we examined buffer sizes $K \in \{20, 50\}$, round trip times $RTT_0 \in \{200 ms, 400 ms\}$ and access rates $r \in \{1$ Mb/s, $2$ Mb/s\}. A representative result is depicted in Figure 1, where the mean transmission delays are presented for the TCP-OnOff model, the TCP-M model and the NS simulation, for the case $r = 2$ Mb/s.

![Image](image_url)

**Fig. 1.** Mean transmission delays for $r = 2$ Mb/s, $\{RTT_0 = 200 ms, K = 20\}$ (left) and $\{RTT_0 = 400 ms, K = 50\}$ (right).

Figure 1 shows that in general the results for the TCP-OnOff model are very good for moderate flow-level loads and although the error grows for a higher load, the global direction of the NS simulation delays is still followed quite well. In other words, even for a high load the On-Off model is able to catch the influence of the buffer size and the round trip time on the transmission delay. Moreover, the On-Off approach clearly outperforms the M/D/1/K approach of [8]. Especially, in the case of a large buffer ($K = 50$) the TCP-M model obviously overestimates the performance of TCP, whereas the TCP-OnOff model yields a quite accurate approximation.

Additional extensive simulations revealed that the above conclusions also hold for other parameter settings. One of the observations we made is that the largest difference with respect to the NS simulation occurs for the setting $\{RTT_0 = 400 ms, K = 20\}$ and $p = 0.9$. For some reason our TCP-OnOff model cannot handle this setting as good as the other investigated values for $RTT_0$ or $K$.

Recall that the maximum advertised window was set to $W_{max} = 1000$ packets to ensure that the maximum throughput is not determined by the maximum window size. To assess the accuracy of
the approximations for scenarios in which the maximum window size does determine the maximum throughput, we have also performed extensive simulations. The results demonstrate that the accuracy of the approximations is comparable to those presented in this section.

5 Conclusions and further research

In this paper we have developed and investigated an enhancement of the integrated packet/flow model (see [8]) for TCP performance analysis. The focus of our research has been on a more accurate description of the packet arrival process of the multiplexed TCP traffic. More specifically, we have studied the effect of adopting an On-Off model on the approximations of the mean file transmission delays. The numerical results show that our enhanced integrated model yields a significantly more accurate fit of the NS simulation outcomes than the TCP-M model of [8]. However, it turns out that for a combination of large RTTs, small buffer sizes and a high load our TCP-OnOff model still underestimates the ability of TCP to control the incoming traffic at the shared link. This inaccuracy needs further investigation.

In addition to studying further model enhancements, extensions of the investigated network are of interest. From a practical point of view it is very interesting to investigate whether the On-Off approach can also be applied in a broader network environment with multiple shared links (cf. Gibbens et al. [5]).

References