Enhancement of an integrated packet/flow model for TCP performance*

R. de Haan[1], J.L. van den Berg[2,3], R.E. Kooij[2] **
R.D. van der Mei[2,4], and A.P. Zwart[1]

[1] Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[2] TNO Telecom,
Brassersplein 2, 2612 CT Delft, The Netherlands
[3] University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
[4] 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 integrated packet/flow model was developed, exploiting the advantages of PS approach at the flow-level, while incorporating the most significant packet-level effects. For that model packet loss and delay were computed using the M/D/1/K queueing model. In this paper we enhance the previous model at the packet-level by modelling TCP flows as On-Off traffic streams. The new analytical model is validated through extensive NS simulations. The results show that the new model leads to significantly better results than the previous integrated packet/flow model.
Keywords: TCP, performance modelling, On-Off model, file download times.

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

* This work is partially funded by the Dutch Ministry of Economic Affairs under the programme 'Technologische Samenwerking ICT Doorbraakprojecten', project TSIT 2031 EQUANET.
** Corresponding author: r.e.kooij@telecom.tno.nl.

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. [1], [2]). 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 [3], [4]).

Recent work of Lassila et al. (see [5]) proposes the integration of the packet- and flow-level model in order to exploit the advantages of both models. In [5] 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 [5] 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 and at packet-level the packet behaviour is represented by a M/D/1/K queueing model. However, in some cases this model is still not very accurate in predicting the mean download times.

In this paper we propose an improvement for the integrated TCP model of [5]. Our contribution mainly concentrates on the packet-level part. We deploy an On-Off model to describe the packet arrival process at the link more accurately. The mean transmission delays following from our integrated approach are verified by extensive NS (see [6]) simulations. The results show that the application of the On-Off model significantly improves the integrated model of [5] 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 [5]). 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 [5] 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 $\frac{1}{\mu_f}$. It is further assumed that the access link rate is limited to r and that all flows have identical RTTs.

2.1 Integrated packet/flow model

The idea of the integrated packet/flow model proposed in [5] 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 [5] 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 [5] is the following (see [3]):

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

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 [5] 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 [5] 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:

$$t_n = min\{nr, \frac{n}{RTT}\sqrt{\frac{2(1-p(t_n))}{p(t_n)}}\} \ , \tag{2}$$

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 [7]), yielding the average number of active flows $E[N]$. Finally, the mean transmission delay $E[D]$ follows by applying Little's law:

$$E[D] = \frac{1}{\lambda} E[N] \ , \tag{3}$$

where λ 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 instants, 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. [8], 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.

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 $d(t_n)$:

$$RTT = RTT_0 + d(t_n) \ , \tag{4}$$

where $d(t_n)$ is the queueing delay following from the On-Off model with n sources active and offered load t_n. In [5] the loss probability p_n and the queueing 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 [9]. Tucker's model contains the following parameters: θ and ν, 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 equal 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 θ and ν.

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:

$$t_{n,indiv} = \frac{t_n \cdot C}{n} \quad . \tag{5}$$

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

$$t_{n,indiv} = \frac{E[\text{On}]}{E[\text{On}] + E[\text{Off}]} = \frac{\theta_n}{\theta_n + \nu_n} \quad , \tag{6}$$

where $E[\text{On}]$ and $E[\text{Off}]$ are the mean duration of an On- and Off-period, respectively, and the subscript n shows the dependence of θ and ν on n. Now θ_n can be expressed as a function of n, t_n, ν_n and C:

$$\theta_n = \frac{\nu_n \cdot t_{n,indiv}}{1 - t_{n,indiv}} = \frac{\nu_n \cdot t_n \cdot C}{n - t_n \cdot C} \quad . \tag{7}$$

This leaves us to set a value for ν_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 ν_n inversely proportional to $\bar{w}$, the mean window size. Further notice that the duration of the On-period also depends on the access rate r, thus ν_n is defined as follows:

$$\nu_n := \frac{r}{\bar{w}} \quad . \tag{8}$$

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 $\bar{w}$. 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 $\bar{w}$, thus:

$$\bar{w} = \sqrt{\frac{2(1-p)}{p}} \quad . \tag{9}$$

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

3.4 Analysis

Tucker [9] provides explicit expressions for the loss rate p_n and the buffer contents distribution $F(x)$ for the On-Off model with exponentially distributed On- and Off-periods. In order to obtain these expressions [9] defines $F_i(x)$ as the equilibrium probability that there are i sources active and the buffer content does not exceed x. These probabilities are found by solving a set of first-order differential equations. To this end, first the eigenvalues z_k and the corresponding eigenvectors $\boldsymbol{\phi}_k$ have to be determined and next the constant coefficients a_k follow from a system of linear equations. The resulting time-independent equilibrium probabilities are of the form:

$$F_i(x) = \sum_{k=0}^{N} e^{z_k x} a_k \phi_{k,i}, \quad 0 < x < m \ , \tag{10}$$

where $\phi_{k,i}$ is the ith element of the eigenvector $\boldsymbol{\phi}_k$. This yields the following expression for the buffer content distribution $F(x)$:

$$F(x) = \sum_{k=0}^{N} F_i(x) \ . \tag{11}$$

After a tedious calculation the mean buffer content $E[L]$ from the distribution $F(x)$ for the case that C is not an integer, can be derived:

$$E[L] = \int_{x=0}^{m} x \cdot dF(x) = (1 - F(m)) \cdot m + \sum_{k=0}^{N} a_k (\mathbf{1}' \boldsymbol{\phi}_k)(m \cdot e^{z_k m} - \frac{1}{z_k} e^{z_k m} + \frac{1}{z_k}) \ , \tag{12}$$

where $\mathbf{1}'$ is the transposed of a vector with all elements equal to one. For the case that C is an integer, (12) should be slightly adjusted. Eventually, we obtain the expression for the mean queueing delay $E[W]$ by applying Little's law:

$$E[W] = \frac{1}{\theta} E[L] \ . \tag{13}$$

Loss will only occur when the buffer is full and the number of active sources i exceeds C. Notice that the amount of loss per unit time equals $i - C$. Introduce the variable u_i as the equilibrium probability that the buffer is full and i sources are active, i.e.

$$u_i = b_i - F_i(m-), \quad 0 \le i \le N \ , \tag{14}$$

where b_i is the is the equilibrium probability of i sources being active and $F_i(m-) = lim_{x \to m} F_i(x)$. This way the loss can be expressed as [9]:

$$p_{loss} = \frac{1}{\alpha \cdot N} \sum_{i=\lceil C \rceil}^{N} (i - C) u_i \ , \tag{15}$$

where $\lceil C \rceil$ stands for the smallest integer larger or equal to C and α is the average fraction of time that a flow is active (i.e. $\alpha = \frac{\theta}{\theta + \nu}$).

Thus, we are able to obtain the required values for the loss and the queueing 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 [5] as TCP-M model. The model outcomes are verified by extensive NS simulations. We have used identical simulation scripts as in [5].

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.

In our experiments we have used the following parameter settings: bottleneck link capacity $c = 10\,Mb/s$, mean file size $\frac{1}{\mu_f} = 1000$ packets, packet size $\frac{1}{\mu_p} =$ 1540 bytes fixed (1500 bytes of data and 40 bytes for the header) and maximum advertised window $W_{max} = 1000$ packets.

First in Subsect. 4.1 we discuss the results for the basic scenario, i.e. access link rate $r = 1\,Mb/s$ and in Subsect. 4.2 we analyse the effect of increasing the access link rate.

4.1 Basic scenario

For the basic scenario we examine the buffer sizes $K \in \{20, 50\}$ and round trip times $RTT_0 \in \{200\,ms, 400\,ms\}$. In Figure 1 the mean transmission delays are presented for the TCP-OnOff model, the TCP-M model and the NS simulation.

Figure 1 shows that in general the results for the TCP-OnOff model are very good for a flow-level load ρ up to 70% 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 [5]. 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.

We observe that the largest difference with respect to the NS simulation occurs for the setting $\{RTT_0 = 400\,ms, K = 20\}$ and $\rho = 0.9$. For some reason our TCP-OnOff model cannot handle this setting as good as the other investigated values for RTT_0 or K. It seems that the relatively large overestimation of the transmission delay is directly related to the relatively high loss rate under these settings. The high loss rate is caused by the combination of a large amount of packets sent each round trip time (i.e. large window size) and a small buffer; as we assume that all packets are sent back-to-back within a round trip time, large flows of packets arrive at the bottleneck link. Thus, the high loss rate follows from the well-known fact that, given a fixed load, larger flows will generate

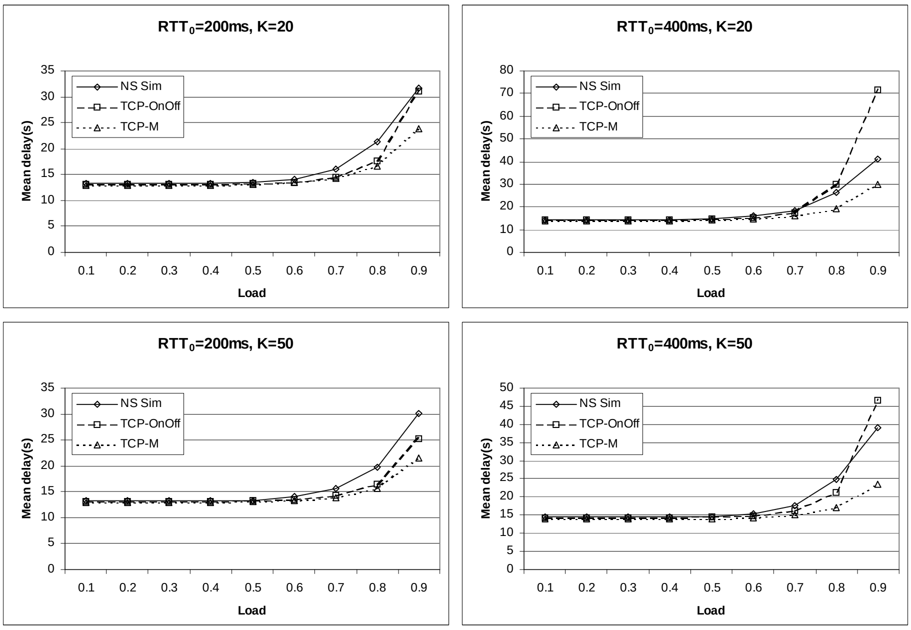

Fig. 1. Mean transmission delays for $r = 1\,Mb/s$, $K \in \{20, 50\}$ and $RTT_0 \in \{200\,ms, 400\,ms\}$.

more losses than small flows (e.g. consider the loss rate for batch arrivals in a M/M/1/K queue).

4.2 High access link rate

In this subsection we present the outcomes for the transmission delays for the case that all parameter settings are as in the basic scenario (see Subsect. 4.1), except that here we study the network model with higher access link rates $r = 2\,Mb/s$. In the case of $r = 2\,Mb/s$, more loss and queueing delay will occur at the bottleneck link due to the burstier packet arrival process, conversely if ρ is small, the mean download times will decrease significantly. Hence, the effect of a decrease or increase of the access bandwidth r on the mean transmission delays is not so obvious at all. In [5] it is shown that the TCP-M does not correctly indicate whether the mean transmission delay will increase or decrease for a high load due to the higher access link rates.

Figure 2 presents the mean transmission delays for an access bandwidth $r = 2\,Mb/s$. The TCP-OnOff model approximates the simulation results very well. A comparison of the transmission delay with the basic scenario (i.e. $r = 1\,Mb/s$) yields that the On-Off model follows the direction of the simulation results, i.e. if for certain parameter settings the simulation shows that the delay increases

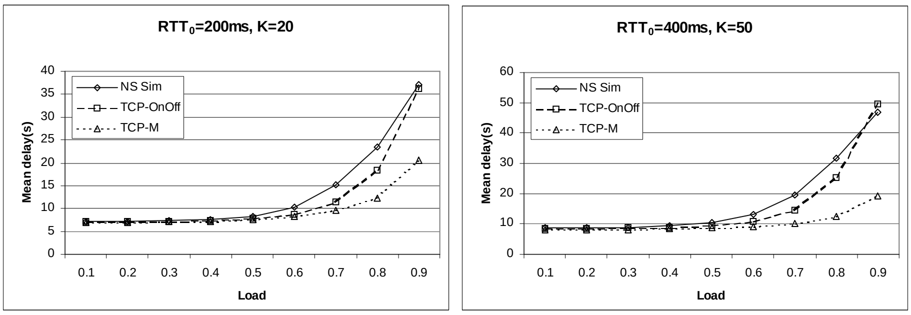

Fig. 2. Mean transmission delays for $r = 2\,Mb/s$, $\{RTT_0 = 200\,ms, K = 20\}$(left) and $\{RTT_0 = 400\,ms, K = 50\}$(right).

for the higher bandwidth, the TCP-OnOff model expects the same. The TCP-M model, however, shows a strong decrease in the mean download times for $\rho = 0.8$ and $\rho = 0.9$ with respect to the $r = 1\,Mb/s$ case, whereas the NS simulation outcomes are higher. Besides, the graphs of Figure 2 show that our TCP-OnOff model estimates the mean download times for $r = 2\,Mb/s$ much better than the TCP-M model.

5 Conclusions and further research

In this paper we have developed and investigated an enhancement of the integrated packet/flow model (see [5]) 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 [5]. In particular, our TCP-OnOff model in contrast to the TCP-M model excellently captures the influence of a higher access link rate. 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. [10]).

References

1. A. Riedl, T. Bauschert, M. Perske and A. Probst, Investigation of the M/G/R Processor Sharing Model for Dimensioning of IP Access Networks with Elastic Traf-

fic, First Polish-German Teletraffic Symposium PGTS, Dresden, 2000.
2. T. Bonald, A. Proutière, G. Régnié and J.W. Roberts, Insensitivity Results in Statistical Bandwidth Sharing, Proc. of the 17th ITC Conference, Salvador, Brazil, December, 2001.
3. F. Kelly, Mathematical Modelling of the Internet, Proc. of 4th International Congress on Industrial and Applied Mathematics, Edinburgh, Scotland, 1999.
4. J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP Throughput: A Simple Model and its Empirical Validation, Proc. of the ACM SIGCOMM '98, Vancouver, Canada, 1998, p303-314.
5. P. Lassila, J.L. van den Berg, M. Mandjes and R.E. Kooij, An Integrated Packet/Flow Model for TCP Performance Analysis, Proc. of the 18th ITC Conference, Berlin, Germany, 2003, p651-660.
6. The Network Simulator - NS2, http://www.isi.edu/nsnam/ns/.
7. J.W. Cohen, The Multiphase Service Network with Generalized Processor Sharing, Acta Informatica 12, 1979, p245-284.
8. D. Anick, D. Mitra and M.M. Sondhi, Stochastic Theory of a Data-Handling System with Multiple Sources, The Bell System Technical Journal, October, 1982, p1871-1894.
9. R.C.F Tucker, Accurate Method for Analysis of a Packet-Speech Multiplexer with Limited Delay, IEEE Trans. on Communications, 1988, 36/4: p479-483.
10. R.J. Gibbens, S.K. Sargood, C. Van Eijl, F.P. Kelly, H. Azmoodeh, R.N. Macfadyen and N.W. Macfadyen, Fixed-Point Models for the End-to-End Performance Analysis, Proc. of the 13th ITC Specialist Seminar: IP Traffic Measurement, Modeling and Management, Monterey, California, USA, 2000.