Performance Analysis of TCP Networks Loaded by Web Traffic

Tamás Éltető Péter Vaderna

Traffic Laboratory, Ericsson Research, H-1117, Budapest, Irinyi J. u. 4-20, Hungary

Sándor Molnár

High Speed Networks Laboratory, Dept. of Telecommunications and Telematics
Budapest University of Technology and Economics
H-1117 Budapest, Magyar tudósok körútja 2., Hungary

Abstract

In this paper a flow level analysis framework is presented for calculating the performance descriptors in TCP networks based on Mean-Value Analysis (MVA). The framework is capable of handling both persistent (ftp-like) and non-persistent (Web-like) TCP connections with arbitrary topology. We can calculate not only average performance descriptors but also detailed flow level traffic and performance characteristics. The proposed algorithm provides large flexibility in the investigated scenarios like existing simulation tools in a fast and memory efficient way. The framework was validated by a simulation study in a test scenario.

Key words: TCP Network Performance, Mean-Value Analysis

---

1 Introduction

The majority of current Internet traffic is carried by the Transmission Control Protocol (TCP). The central role of TCP called for intensive research on understanding TCP dynamics and developing planning and dimensioning methods for TCP networks. Most of these studies investigating the performance of TCP networks are using the assumption of persistent sources which always have data

---

Email addresses: `tamas.elteto@eth.ericsson.se` (Tamás Éltető), `peter.vaderna@eth.ericsson.se` (Péter Vaderna), `molnar@ttt-atm.ttt.bme.hu` (Sándor Molnár).

Preprint submitted to Elsevier Science *16 May 2003*

to send. However, the most popular application in the current Internet is the World Wide Web (WWW). The WWW is based on the HTTP protocol that mainly uses non-persistent file transfers to retrieve Web pages. Therefore there is a need for an analysis framework including also non-persistent TCP sources and this is the goal of this work.

There are several related papers in the field of TCP modelling and performance analysis of TCP networks. The basic model of the TCP was developed and the well-known inverse square root law was investigated e.g. in [11,13]. A number of variants of these basic models were also developed by relaxing several assumptions of the original scenario, e.g. in [6]. Our framework has the flexibility that one can choose from these developments based on the specific environment and fit the chosen TCP model into our framework.

In contrast to several earlier studies we deal with *closed-loop modelling principle* instead of the open-loop methodology which was shown to be inadequate for proper TCP modelling, see e.g. [2]. The main cause of the problems related to open-loop modelling is the TCP feedback mechanism. It is well-known that TCP reduces its sending rate under lossy network conditions. On the other hand, the adaptivity of TCP makes it possible that it copies the statistical properties of the background traffic it is mixed with. Moreover, the TCP mechanism can also adapt to self-similar fluctuations on several time scales [19].

In our work we consider a closed queueing network which integrates the self-clocking mechanism inherent in TCP networks. As a solution method we use the Mean-Value Analysis (MVA) which is a well developed technique [15]. In the literature there are some related papers that use the fixed-point methods [1,3,5,9,16,18]. For example, [5] considers the TCP flow behaviour with active queue management. [16] presents similar results for TCP network performance. However, the main difference from our framework is that we consider not only persistent but also non-persistent sources and we describe the number of current TCP flows by a separate Markov-based submodel. We note that the possibility of using the MVA was already suggested in [16]. [7] is also based on the concept of closed queueing networks for the description of the TCP state machine of TCP-Tahoe, while we use it to describe the IP network performance. The results in [9] consider also non-persistent TCP sources but this analysis is based on the Engset model.

The main contribution of the paper is that it provides a practical analysis framework for TCP network performance analysis with the following main features:

(1) it has a *TCP submodel* which can incorporate any of the advanced TCP models;
(2) it has a *network submodel* which is based on a closed-loop modelling

principle and uses MVA;
(3) it has a *flow submodel* which captures the dynamics of parallel non-persistent TCP sources based on a Markov chain;
(4) it does not assume any particular network topology;
(5) it can provide both average performance descriptors (e.g. queue length, round-trip time, throughput, download time, etc.) and detailed flow level traffic and performance characteristics (e.g. approximate distributions of the above listed characteristics).

2 The Network Topology

Several recent publications dealing with the theoretical analysis of TCP traffic discuss arbitrary topology networks analysed by fixed point methods (see e.g. [1,3,16,18]). All of these models obtain an average value for the number of parallel connections, average round-trip times and in this way the average download times for short transfers and average throughputs for persistent transfers.

There are other models that obtain more detailed results on the number of parallel connections (e.g. [4,9]) but in these cases, the network topology is not arbitrary.

The proposed method can handle theoretically arbitrary topology networks and is able to obtain not only average performance descriptors but also more detailed results (e.g. distributions). However, we note that a more complex network topology may imply increased computational requirements determined by the MVA solution techniqe.

3 An Analytic Framework

In this section we describe our framework (see Figure 1). The main performance descriptors of interest here are the average queue lengths and the average round-trip times (RTTs). Application level performance descriptors like average download time or average throughput can be obtained directly from the average RTT using recent results [1,3,16] for short and persistent TCP connections.

The model consists of three submodels, the first of which is responsible for the TCP behaviour, the second one describes the network response and the third one deals with the dynamics in the system on flow level.

The *TCP submodel* describes the TCP's response given certain network conditions. This submodel computes the average number of packets kept in the network by one TCP connection and the average download time in round-trip units.

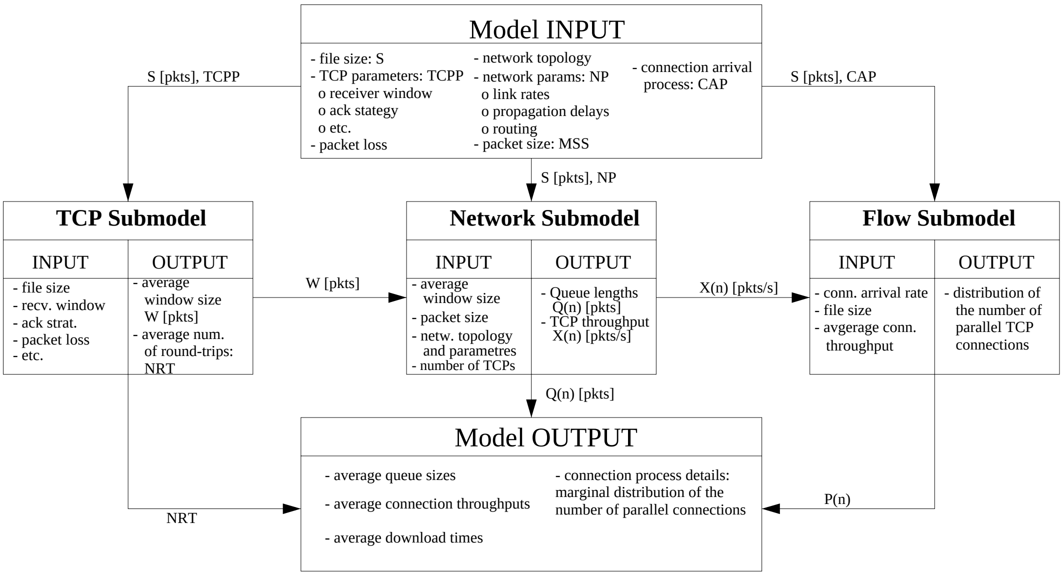

Fig. 1. The analysis framework

The *network submodel* computes the network performance (e.g. RTT, throughput) given the number of packets circulating in an arbitrary topology. This submodel uses the MVA technique.

The *flow submodel* deals with the dynamics of the network by modelling the constantly changing load according to a Markovian process.

3.1 TCP submodel

The TCP submodel provides the average offered load to the system given the the packet loss probability and other parametres. Several publications revealed that the average window size of a TCP source is proportional to the inverse square root of the loss probability e.g. in [13]. This *inverse square root law* was first proved for the case where some restrictive assumptions hold (e.g. low loss rate, persistent sources, neglecting slow-start, etc.). Further research demonstrated that by relaxing these assumptions makes the exact formula more complex, but the inverse square root law still holds. The appropriate TCP model can be chosen from these results depending on the investigated scenario (e.g. we chose the one developed in [6]). The TCP Reno version was used in the analysis and in the simulations.

The TCP submodel gives the average number of round-trip times needed to download a file. The application level performance descriptors like the file download time in *sec* can be obtained as a product of this output and the average RTT. Finally, the throughputs of the persistent TCP connections are the side-results of the method.

3.2 Network submodel

The network submodel consists of a closed queuing network of IP packets with arbitrary network topology analyzed by the MVA [10,15]. The motivation that MVA was chosen for the network submodel is that it can model appropriately the self-clocking mechanism of TCP. The MVA method is the iteration of equations on the average waiting time in queues (average response time), the average throughput and the average queue length. The input of this submodel is the average window size (W [packets]), network parametres and the number of parallel TCP connections. The network submodel computes the conditional average performance descriptors based on the above input parametres.

Denote by n the number of parallel connections in the system. The method below is performed on a set of possible values for n. The number of packets circulating in the closed-loop system is $n \cdot W$ packets. Denote $R_k(n)$ the average waiting time upon a packet arrival in queue k. $R_k(n)$ is related to the average queue length Q_k depending on the type of the MVA being used ($f(Q_k)$). (The theoretical explanation can be found e.g. in [15].)

$$R_k(n) = S_k(1 + f(Q_k)), \tag{1}$$

where S_k is the average service demand of one packet. If all the average response times are given, then the average throughput $X(i)$ is

$$X(n) = \frac{n \cdot W}{\sum_k R_k(n)}. \tag{2}$$

The average queue lengths can be computed from the average response time and the average throughput by

$$Q_k(n) = X(n) R_k(n). \tag{3}$$

If the number of flows is large, then the original MVA method does not scale well. However, there exist approximate MVA algorithms that provide good estimations with significantly reduced computational complexity [10].

3.3 Flow submodel

Our flow submodel describes the changing number of TCP connections in progress. The model consists of a Markov-chain which is an embedded Markov-chain at packet departure instants, where the states are the number of parallel TCPs. It can be seen that being the last packet at a departure instant is an independent event having the same probability for all states ($p = 1 - e^{-1/S}$) if the file size is exponential (with average S packets). The TCP connection arrivals are modelled by the Poisson arrival process and the expected time between two consecutive packet departures from a certain service point is

$$E(T_n) = \frac{1}{X(n)}, \tag{4}$$

where n is the number of TCPs and $X(n)$ is the conditional average throughput obtained from the network submodel. The probability that j new TCPs arrive given n ongoing TCPs is

$$p(n, j) = \frac{(\lambda E(T_n))^j}{j!} e^{-\lambda E(T_n)}, \tag{5}$$

where λ is the connection arrival rate.

Stationary probabilities of the Markov chain can be computed by various Markov-chain solution techniques [17].

Finally the average queue lengths can be obtained from the conditional averages by taking their weighted sum with the steady-state probabilities as weights.

$$Q_k = \sum_{n=0}^{\infty} Q_k(n) P(n). \tag{6}$$

Note that though the properties of the Poisson flow arrival process were used here, there are results e.g. [12] showing that this is not always in line with actual measurements. However, the arrival process can be generalised to a Markov-Modulated Poisson Process process by extending the state-space of the above Markov-chain by taking into consideration the state transitions of the MMPP together with the transitions related to TCP arrivals/departures.

4 Test Results

The analytical results was tested by simulations. The simulation tool was the IP version of the Plasma platform (Planning Algorithms and Simulation for Network Management, for details see [8]). Plasma IP is a packet-level simulation tool including an implementation of the the Reno version of the TCP protocol developed by the HSN Lab at Budapest University of Technology and Economics and Ericsson Traffic Lab.

The topology in our example network contains three FIFO queues and a packet delay. There are two traffic flows shaped by queue A and B and multiplexed in a common queue (see Figure 2). This is a simple illustrative example, but we note that our method was also tested in different scenarios with similar accuracy.

The traffic flows contain two types of TCP transfers, where the connections of the same type have common sources and destinations. The goal of the proposed method is to predict the average queue lengths and average packet round-trip times that develop in the IP network.

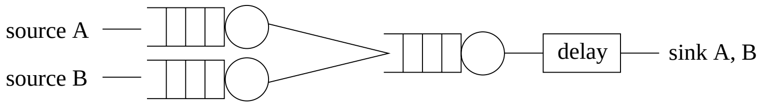

Fig. 2. Network topology example

The connection arrival intensities and the average file sizes are different for flows A and B. The connection arrival process is a Poisson process in the present network example[1] . The file sizes are assumed to be exponential [2] .

The simulations were repeated with different server utilisation levels. The parametres of the investigated network were the following:

- The bandwidth of the common link was 144 x 1/utilisation kbps[3] , the bandwidth of source A's link was 80 x 1/util. kbps and the bandwidth of source B's link was 64 x 1/util. kbps,
- 0.5 s fixed packet delay,
- the packet size was 1500 bytes,
- the agent used TCP Reno with 10 packets receiver window. No Nagle algorithm and no delayed acknowledgement were used.

The average file sizes in flows A and B were 20 and 40 kbyte, respectively. The offered load of the TCP traffic was 80 and 64 kbps.

Figure 3 shows the distributions of the number of parallel TCP connections in the simulation and in our computations. The computed distributions well follow the empirical ones.

---

[1] There are results showing that the Poisson model for connection-level arrival process is not always appropriate, see e.g. [12]. This was used for the sake of computational simplicity, nevertheless this is not a critical assumption as it was discussed in Section 3.2.

[2] Contrary to the fact that a number of papers showed that these distributions have heavy-tails, see e.g. [14], our modelling approach is that the body of a file size distribution can be approximated by the mixture of exponential distributions and the files in the tail (the extremely large files) are simply considered as persistent file transfers.

[3] For example, in the case of 0.63, 0.73, 0.64 server utilisations, the corresponding link rates were 230, 110, 100 kbps.

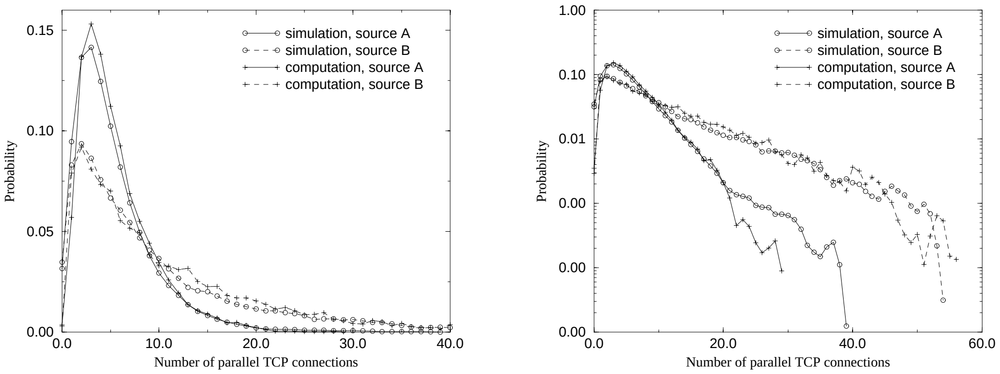

Fig. 3. The distributions of the number of parallel TCP connections (lin-lin and lin-log scale)

To give a more comprehensive validation, two performance descriptors were computed and logged in simulations[4] in 10 test cases:

- the packet round-trip delays on the routes of the two source-sink pairs (route A and B),
- the file download times of the files transmitted between the source-sink pairs.

Table 1 shows the parameter set on which the comparison was done. Table 2 compares the RTT results from computations and simulations. Table 3 compares the computed average download times and the results of the simulations.

Table 1

The utilisations in the three servers in 10 test cases

Test	1	2	3	4	5	6	7	8	9	10
Utilisation in queue A	0.73	0.73	0.73	0.73	0.73	0.80	0.85	0.73	0.73	0.73
Utilisation in queue B	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.75	0.80	0.85
Utilisation in queue AB	0.63	0.75	0.80	0.85	0.90	0.63	0.63	0.63	0.63	0.63

Table 2

The comparison of the computed average RTT and simulation results

Test	1	2	3	4	5	6	7	8	9	10
RTT A computed	1.6	2.1	2.7	3.7	6.2	2.5	3.9	1.7	1.7	1.6
RTT A exp.	1.8	2.0	2.5	3.2	8.6	2.4	4.8	1.7	1.7	1.8
RTT A non-exp.	1.6	2.2	2.1	3.1	4.5	2.3	5.3	1.7	1.7	1.6
RTT B computed	2.3	2.3	2.8	3.6	6.1	2.0	2.0	3.5	4.3	6.6
RTT B exp.	2.1	2.2	2.6	3.4	8.3	2.1	2.0	3.5	4.4	6.5
RTT B non-exp.	2.1	1.8	2.6	3.2	4.4	2.0	1.9	3.2	5.9	5.6

Larger differences occur when the utilisation becomes large on the links, which might be due to the high variance of the queue length and certain TCP effects

---

[4] Two different models were used in the simulations. The first one assumes exponential packet sizes (the model is strongly related to a BCMP-type network). The second case uses the packet sizes determined by the TCP stack.

Table 3
The comparison of the predicted download times and simulation results

Test	1	2	3	4	5	6	7	8	9	10
T_d^A computed	9.3	12.1	15.6	21.4	35.8	14.5	22.5	9.8	9.8	9.3
T_d^A exp.	10.3	13.5	13.5	20.5	44.1	14.3	26.5	10.7	11.6	10.5
T_d^A non-exp.	10.7	11.5	15.9	18.5	30.4	13.3	27.7	10.6	11.1	9.8
T_d^B computed	15.2	15.2	18.5	21.4	40.3	13.2	13.2	23.1	28.4	43.6
T_d^B exp.	16.3	16.2	14.9	26.0	55.5	13.9	14.8	22.6	29.9	54.3
T_d^B non-exp.	15.0	14.3	22.2	23.0	39.4	13.9	14.3	26.5	27.8	33.8

that were not studied in detail in the present paper. The average RTT predictions therefore estimate the simulated averages with mostly less than 5 %-15 % error.

5 Conclusions and Future Work

In this paper we presented a performance modelling framework based on a closed-queuing network analysis which is able to describe the performance of a TCP network both with persistent and also with changing number of non-persistent TCP connections.

Based on the present work it is possible to estimate a number of average network descriptors such as the average queue length and the average RTT. Application level performance descriptors such as average download time can also be obtained from the output of the proposed method.

The results were verified by simulations not only for the average RTTs but also for the distribution of the number of parallel TCP connections and for the average queue lengths.

The strict assumptions in the present model, namely the Poisson connection arrival process and exponential file size can also be relaxed at the expense of increased computational complexity.

More subtle application performance descriptors can also be obtained based on a refined Markovian approach for the flow submodel. Instead of the average file download time, for example, it is possible to compute the WWW page download time when multiple file transfers are allowed.

References

[1] E. Alessio, M. Garetto, R. Lo Cigno, M. Meo, M. Ajmone Marsan, “Analytical Estimation of Completion Times of Mixed NewReno and Tahoe TCP Connections over Single and Multiple Bottleneck Networks”, Proceedings of IEEE Globecom 2001, San Antonio, Texas, USA, 2001.

[2] A. Arvidsson and P. Karlsson, ”On traffic models for TCP/IP”, Proceedings of ITC16, Edinburgh, U.K., 1999.

[3] A. Arvidsson, A. Krzesinski, “A Model of a TCP Link”, Proceedings of 15th ITC Specialist Seminar, Internet Traffic Engineering and Traffic Management, Wuerzburg, Germany 2002.

[4] S. Ben Fredj, T. Bonalds, A. Prutiere, G. Gegnie, and J. Roberts, “Statistical bandwidth sharing: a study of congestion at Flow level”, In Proceedings of ACM SIGCOMM 2001, San Diego, California, USA, 2001.

[5] T. Bu, D. Towsley, ”Fixed Point Approximations for TCP Behaviour in AQM Network”, Proc. of ACM SIGMETRICS, June 16-20, Cambridge, USA, 2001.

[6] N. Cardwell, S. Savage, T. Anderson, “Modeling TCP latency”, Proceedings of Infocom 2000, Tel-Aviv, Israel, 2000.

[7] M. Garetto, R. Lo Cigno, M. Meo, M. A. Marsan, “A Detailed and Accurate Closed queuing Network Model of Many Interacting TCP Flows”, Proceedings of Infocom 2001, Anchorage, Alaska, USA, April 22-26, 2001.

[8] Z. Haraszti, I. Dahlqvist, A. Faragó, T. Henk, “PLASMA - A Tool for ATM Network Operations”, Proceedings of the 15th International Switching Symposium (ISS), Berlin, 1995.

[9] D. P. Heyman, T. V. Lakshman and Arnold L. Neidhardt, “A new method for analysing feedback-based protocols with applications to engineering Web traffic over the Internet”, ACM Sigmetrics 1997, Seattle, Washington, USA, 1997.

[10] E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik, “Quantitative System Performance Computer System Analysis Using Queueing Network Models”, Prentice-Hall, Inc., 1984.

[11] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. ”The macroscopic behaviour of the TCP congestion avoidance algorithm”, Computer Communication Review, vol. 27, pp. 67-82, July 1997.

[12] C. Nuzman, W. Sweldens, A. Weiss, “A Compound model for TCP connection arrivals: Empirical study with a general framework”, Proceedings of ITC 2001, Brazil, Dec 2001.

[13] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, “Modeling TCP throughput: A simple model and its empirical validation”, ACM Computer Communication Review, 28(4), September 1998, pp. 303-314.

[14] V. Paxson and S. Floyd, ”Wide-area traffic: The failure of Poisson modeling”, IEEE/ACM Trans. Networking, 3(3):226-255, 1995.

[15] M. Reiser, S. S. Lavenberg, “Mean Value Analysis of Closed Multichain Queuing Networks“, Journal of the Association for Computing Machinery, Vol 27, No 2, April 1980, pp 313-322.

[16] M. Roughan, A. Erramilli and D. Veitch, “Network Performance for TCP Networks Part I: Persistent Sources”, Proceedings of ITC’17, Brasil, September 2001.

[17] W. J. Stewart, “Introduction to the Numberical Solution of Markov Chains”, Princeton University Press, 1994

[18] A. De Vendictis, A. Baiocchi, A. Monticelli, “Simple Models and their Limits for TCP/IP Network Analysis and Dimensioning”, in Proc. of IEEE ICC2002, April 2002.

[19] A. Veres, Zs. Kenesi, S. Molnár, G. Vattay, “On the Propagation of Long-Range Dependence in the Internet”, ACM Computer Communication Review, Vol. 30, No.4, pp. 243-254, October, 2000.