The effect of router buffer size on the TCP performance

K.E. Avrachenkov[*], U. Ayesta[*,†] E. Altman[*,‡], P.Nain[*], C. Barakat[*,§]

December 20, 2001

Abstract

In this paper we study the effect of the IP router buffer size on the sending rate, the goodput and the latency of a TCP connection. We analyse short TCP transfers as well as persistent TCP connections. Analytic results obtained by fixed point approach is compared with the results obtained by NS simulator.

Key words. TCP/IP, Fixed Point Approach, the buffer size of IP Router.

1 Introduction

The effect of the IP router buffer size on the TCP performance is analysed in this paper. We consider two types of TCP connections: persistent TCP connections that have always data to send and short TCP transfers of small files. A typical example of the first type is an FTP transfer and a typical example of the second type is an HTTP transfer. In the first case we are interested in the effect of the buffer size on the average values of the TCP sending rate and goodput, and in the second case we are interested in the effect of the buffer size on the latency of the file transfer. We carry out the analysis using the Fixed Point Approach (FPA) developed in [1, 3]. The main idea behind FPA for the TCP/IP network is to combine a model for the IP network at the packet level with a model for the TCP connection performance based on some given packet loss process. In [1] we applied FPA to the modeling of TCP/IP network in the case of large bandwidth-delay products, that is when the queueing delay may be neglected. In contrast, here we want to study the effect of the queueing delay on the performance of TCP sessions. In [3], the authors analyse the case of RED buffers. Here we propose a model for drop tail buffer, which is still the most common type of buffers in IP networks. In particular, we answer the question if there is an optimal size of the buffer. In this paper, we restrict ourselves to the case of one bottleneck node. However, our approach can easily be generalized to a more sophisticated topologies. We confirm all our theoretic findings with the help of NS simulations.

2 Persistent TCP connections

Let us consider N persistent TCP connections that are going through the same bottleneck link (see Figure 1) of capacity C. The links leading to the bottleneck link may have different

[*] INRIA Sophia Antipolis, 2004 route des Lucioles, B.P.93, 06902, Sophia Antipolis Cedex, France. e-mail: {k.avrachenkov, uayesta, altman, nain, cbarakat}@sophia.inria.fr

[†] The work of this author was supported by grant of Alcatel Space Industries during his internship in INRIA. Currently the author is with France Telecom R&D, 06921 Sophia-Antipolis, France

[‡] C.E.S.I.M.O. Centro de Simulacion y Modelos Universidad de Los Andes (U.L.A.) Facultad de Ingeneria, Av. Tulio Febres Codero, Merida, Venezuela, e-mail:altman@ing.ula.ve

[§] EPFL-DSC-ICA, INN-Ecublens, CH-1015 Lausanne, Switzerland, e-mail: Chadi.Barakat@epfl.ch

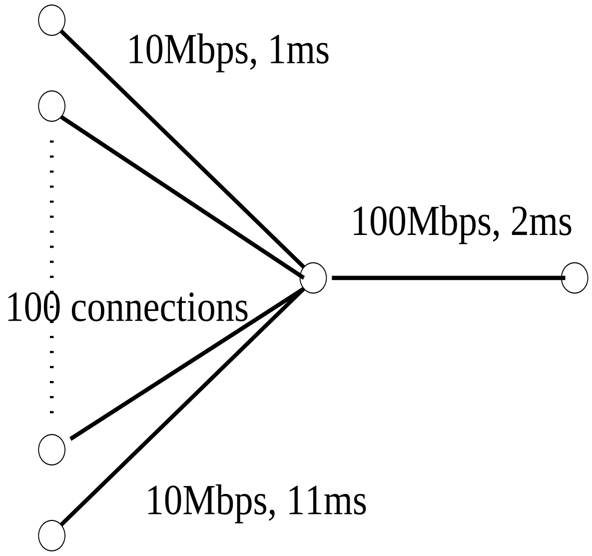

Figure 1: N persistent TCP connection sharing a single bottleneck link.

propagation delays and their capacities are sufficiently large so that only the bottleneck link experiences congestion.

Denote by RTT_i the average Round Trip Time of the i-th connection. Given $RTT_i, i = 1, ..., N$ and the packet loss probability p on the connection path which in the case of a single bottleneck is the same for all connections, the average TCP sending rate of a persistent connection can be calculated by the following formula [5]

$$T_i(p, RTT_i) = \begin{cases} MSS_i \frac{\frac{1-p}{p} + W(p) + Q(p, W(p))}{RTT_i(\frac{b_i}{2}W(p)+1) + \frac{Q(p,W(p))F(p)T_0^i}{1-p}}, & \text{if} \quad W(p) < W_{max}^i, \\ MSS_i \frac{\frac{1-p}{p} + W_{max}^i + Q(p, W_{max}^i)}{RTT_i(\frac{b_i}{8}W_{max}^i + \frac{1-p}{pW_{max}^i} + 2) + \frac{Q(p,W_{max}^i)F(p)T_0^i}{1-p}}, & \text{otherwise,} \end{cases} \tag{1}$$

where

$$W(p) = 2/3 + 2\sqrt{(1-p)/(3p) + 1/9},$$

$$Q(p, w) = \min\{1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{w-3}))}{1-(1-p)^w}\},$$

$$F(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6,$$

and where MSS_i is the Maximal Segment Size, W_{max}^i is the maximal receiver window, b_i is the number of packets acknowledged by an ACK, and T_0^i is the basic timeout duration. Then, the load on the bottleneck link is given by

$$\rho = \frac{1}{C} \sum_{i=1}^{N} T_i(p, RTT_i). \tag{2}$$

Thus, we have described the TCP part of our FPA model. Next we consider a model for the IP layer. To express the packet loss probability p and the queueing delay in terms of the load ρ and the IP router buffer size K, we use the classical M/M/1/K queueing model

$$p = \rho^K \frac{1-\rho}{1-\rho^{K+1}}, \tag{3}$$

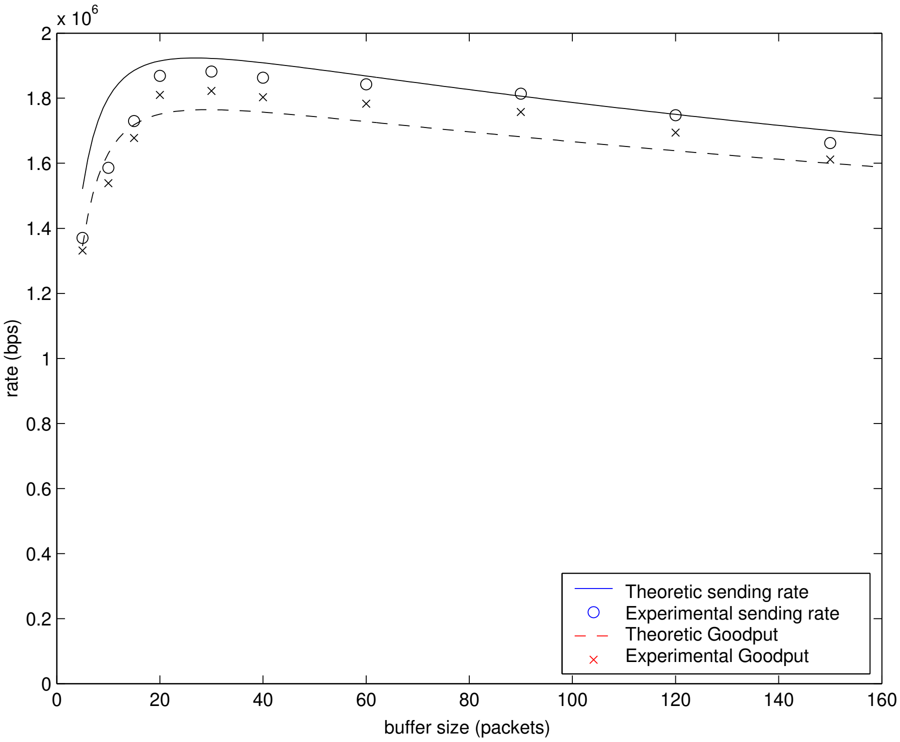

Figure 2: TCP connection with the smallest propagation delay.

$$RTT_i = 2 * d_i + \frac{MSS}{C}\left[\frac{1}{1-\rho} - K\frac{\rho^K}{1-\rho^K}\right], \quad (4)$$

where d_i is the propagation delay of the i-th connection. In the above expression we assign to MSS the average maximal segment size. Substituting (3) and (4) into equation (2), we get the following nonlinear equation for ρ.

$$\rho = \frac{1}{C}\sum_{i=1}^{N} T_i\left(\rho^K \frac{1-\rho}{1-\rho^{K+1}}, d_i + \frac{MSS}{C}\left[\frac{1}{1-\rho} - K\frac{\rho^K}{1-\rho^K}\right]\right)$$

The above equation can be solved by any numerical method for the solution of nonlinear equations. Once the value of ρ is determined, we can compute p, RTT_i and the sending rate T_i by formulae (3), (4) and (1), respectively. The goodput is calculated by the formula

$$G_i = (1-p) * T_i(p, RTT_i). \quad (5)$$

In our numerical experiments with NS simulator [4] we set the capacity of the bottleneck link as 100Mbps and its propagation delay as 2ms. Each of 100 links which lead to the bottleneck link has the capacity of 10Mbps. The propagation delays of these links are uniformly distributed between 1ms and 11ms. In our simulations we use the New Reno TCP version. In Figure 2,3 we plot the sending rate and the goodput for the TCP connections corresponding to the shortest and to the longest propagation delays.

3 Short TCP connections

Let us now consider short TCP transfers passing through a single bottleneck with the capacity C. This scenario corresponds to the HTTP file transfers. We suppose that TCP sessions

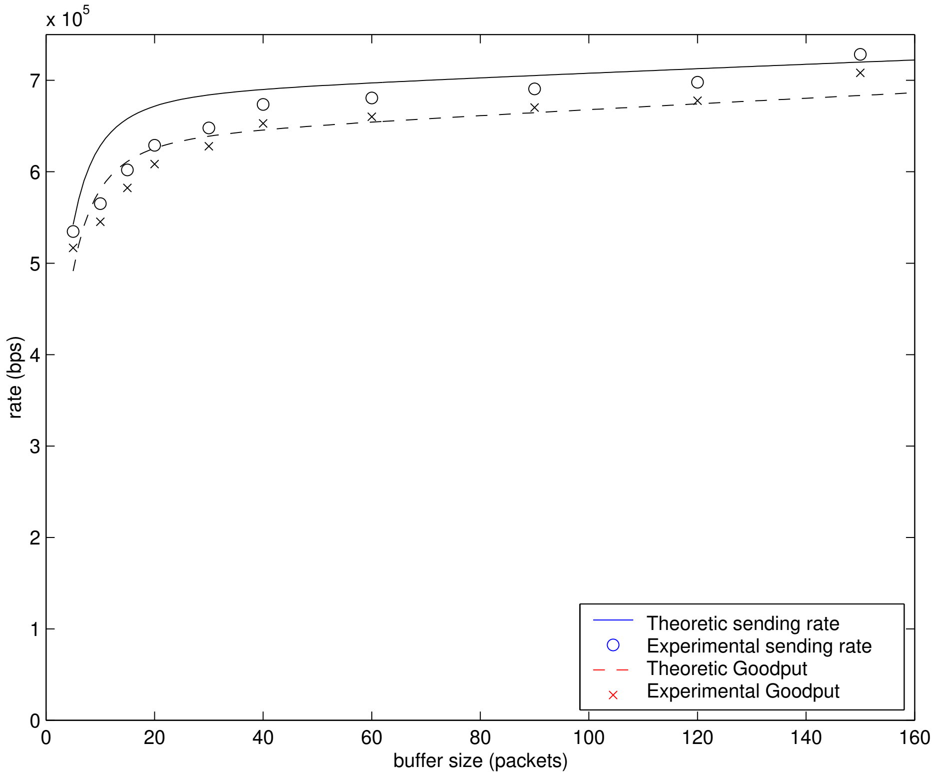

Figure 3: TCP connection with the largest propagation delay.

arrive according to a Poisson process with rate λ. The load can be defined as

$$\rho_0 = \frac{\lambda E[doc.size]}{C},$$

where $E[doc.size]$ is the average document size. For instance, the average size of documents transfered by HTTP is 10Kbytes. The fixed point formulation corresponding to the above setting can be written as follows

$$\rho = \frac{\rho_0}{1-p},$$

$$p = \rho^K \frac{1-\rho}{1-\rho^{K+1}}.$$

We note that the first equation in the above system is taken from [2]. This system can be solved by using fixed point iterations

$$\rho_i = \frac{\rho_0}{1-p_i},$$

$$p_{i+1} = \rho_i^K \frac{1-\rho_i}{1-\rho_i^{K+1}},$$

for $i = 1, 2, ...$ Once we have obtained p and ρ, we can calculate the RTT corresponding to the propagation delay d. Namely,

$$RTT = 2 * d + \frac{MSS}{C} \left[\frac{1}{1-\rho} - K \frac{\rho^K}{1-\rho^K} \right].$$

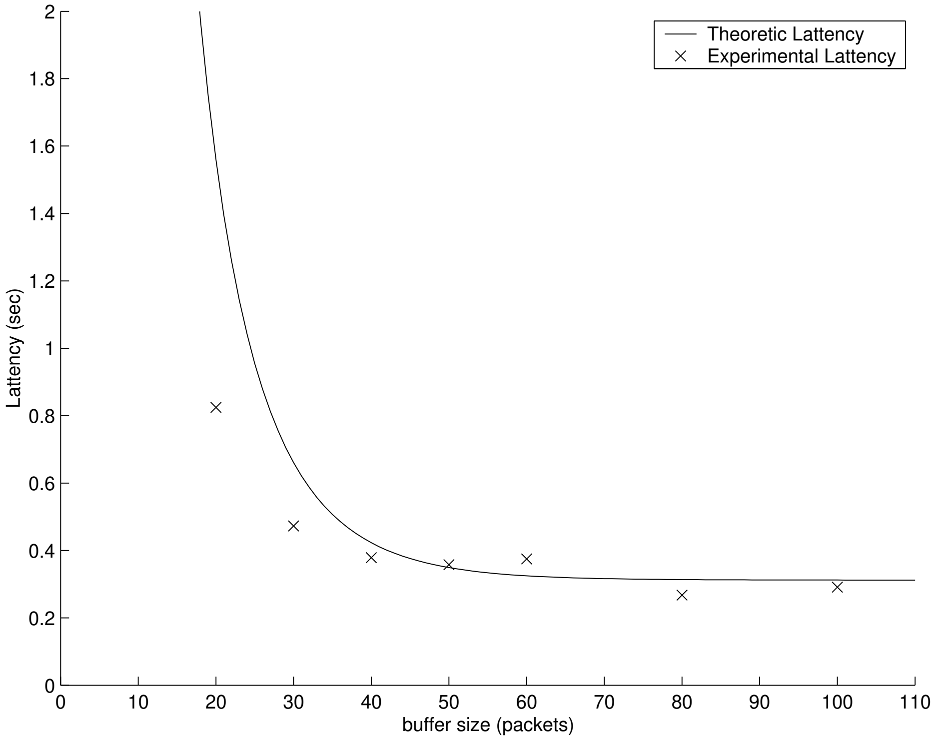

Figure 4: The latency for the document of 20 packets.

Finally, the following formula taken from [6] can be used to estimate the average transfer latency for a document of m packets

$$L(m) = RTT\left(\log_{1.57}(m) + f(p, RTT)m + 4p\log_{1.57}(m) + 20p + \frac{10 + 3RTT}{4(1-p)W_{max}\sqrt{W_{max}}}m\right), \tag{6}$$

where

$$f(p, RTT) = \frac{2.32(2p + 4p^2 + 16p^3)}{(1 + RTT)^3}N + \frac{1 + p}{RTT10^3}.$$

For NS simulations we have chosen the following parameters $E[doc.size] = 10Kbytes$, $\lambda = 225sessions/s$ and $C = 20Mbps$. This set of parameters gives the link load of 90%. In Figure 4 we plot the relation between the expected document transfer latency and the buffer size for the document of 10Kbytes (or 20 IP pakets, 500bytes each).

4 Conclusions and future work

From the theoretical analysis and the experiments with NS we conclude that typically a persistent TCP connection crossing a bottleneck link has poor sending rate and goodput in the cases of small and large buffer sizes. The explanation for the first case is that small buffers induce many packet losses and the explanation for the second case is that queueing delays and hence round trip times become too large. Consequently, for a given TCP connection there is an optimal value of the buffer size for the bottleneck IP router. However, this optimal value is different for each TCP connection since it depends on the value of the propagation delay (see Figures 2 and 3). One can also see that the deterioration in sending rate and goodput for large buffers is not so significant as in the case of small buffers. Thus, in order for the system to be robust, we suggest to set the buffer size rather larger than smaller. Actually, this recommendation applies equally well to the case of short TCP transfers. As demonstrated in Figure 4, with the increase of the bottleneck buffer size, the latency of the file transfer decreases. It seems that there is no optimal value for the IP router buffer size if we consider only short TCP transfers.

We also note that the fairness always improves with increasing buffer size, as the share of bandwidth of the connections with large propagation delays increases and those of short propagation delays decreases. This can be explained by the increase of the average queueing delay which is the same for all TCP connections sharing the same bottleneck link.

The experimental results that we obtained were all with New-Reno TCP version. When using Reno we obtained in our NS experiments around 30sending rates. This can be explained by the fact, which we have observed through the traces, that Reno has many more time-out losses. Moreover, we observed that Reno often had several losses in the same window and recovering from these resulted in several consecutive decreases of the window size. In contrast, New-Reno is able to recover well from more than one consecutive loss within an RTT.

Now let us point out several future research directions. The first obvious future research direction is to investigate more sophisticated network topologies with several bottleneck links. The present approach seems to allow this generalization. Next we can conclude from Figure 4 that formula (6) does not work well in the case of large packet losses. Thus, one needs to obtain a more exact formula for the latency of TCP based file transfer.

References

[1] E. Altman, K.E. Avrachenkov and C. Barakat, "TCP network calculus: The case of large delay-bandwidth product", accepted for *IEEE INFOCOM 2002* conference, an electronic version is available at http://www-sop.inria.fr/mistral/personnel/K.Avrachenkov/me.html.

[2] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, Ph. Owezarski, "A flow-based model for Internet backbone traffic", EPFL Technical Report DSC/01/44.

[3] T. Bu and D. Towsley, "Fixed point approximation for TCP behaviour in an AQM network", in Proceedings of SIGMETRICS'2001 conference.

[4] Network Simulator, Ver.2, (NS-2), available at http://www-mash.cs.berkeley.edu/ns.

[5] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", in Proceedings of *ACM SIGCOMM'1998* conference.

[6] B. Sikdar, S. Kalyanaraman and K. S. Vastola, "An Integrated Model for the Latency and Steady-State Throughput of TCP Connections", *Performance Evaluation*, vol. 46, no. 2-3, pp. 139-154, September 2001.