Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Optimal choice of the buffer size in the Internet routers

Konstantin Avrachenkov
INRIA Sophia Antipolis, France
e-mail: k.avrachenkov@sophia.inria.fr

Urtzi Ayesta
CWI, The Netherlands
e-mail: ayesta@cwi.nl

Alexei Piunovskiy
University of Liverpool, UK
e-mail: piunov@liverpool.ac.uk

Abstract— **We study an optimal choice of the buffer size in the Internet routers. The objective is to determine the minimum value of the buffer size required in order to fully utilize the link capacity. There are some empirical rules for the choice of the buffer size. The most known rule of thumb states that the buffer length should be set to the bandwidth delay product of the network, i.e., the product between the average round trip time in the network and the capacity of the bottleneck link. Several recent works have suggested that as a consequence of the traffic aggregation, the buffer size should be set to smaller values.**

In this paper we propose an analytical framework for the optimal choice of the router buffer size. We formulate this problem as a multi-criteria optimization problem, in which the Lagrange function corresponds to a linear combination of the average sending rate and average delay in the queue. The solution to this optimization problem provides further evidence that indeed the buffer size should be reduced in the presence of traffic aggregation. Furthermore, our result states that the minimum required buffer is smaller than what previous studies suggested. Our analytical results are confirmed by simulations performed with the NS simulator.

I. INTRODUCTION

Most traffic in the Internet is governed by TCP/IP protocol [1], [9]. Data packets of an Internet connection travel from a source node to a destination node via a series of routers. Some routers, particularly edge routers, experience periods of congestion when packets spend a non-negligible time waiting in the router buffers to be transmitted over the next hop. TCP protocol tries to adjust the sending rate of a source to match the available bandwidth along the path. During the principle Congestion Avoidance phase TCP uses Additive Increase Multiplicative Decrease (AIMD) control scheme. In the absence of congestion signals from the network TCP increases sending rate linearly in time, and upon the reception of a congestion signal TCP reduces the sending rate by a multiplicative factor. Congestion signals can be either packet losses or Explicit Congestion Notifications (ECN) [15]. At the present state of the Internet, nearly all congestion signals are generated by packet losses. Packets can be dropped either when the router buffer is full or when Active Queue Management (AQM) scheme is employed [7]. Given an ambiguity in the choice of the AQM parameters [5], [10], so far AQM is rarely used in practice. On the other hand, in the basic Drop Tail routers, the buffer size is the only one parameter to tune apart of the router capacity. In fact, the buffer size is one of few parameters of the TCP/IP network that can be managed by network operators. This makes the choice of the router buffer size a very important problem in the TCP/IP network design.

The first proposed rule of thumb for the choice of the router buffer size was to choose the buffer size equal to the Bandwidth-Delay Product (BDP) of the outgoing link [17]. This recommendation is based on very approximative considerations and it can be justified only when a router is saturated with a single long-lived TCP connection. The next apparent question to ask was how one should set the buffer size in the case of several competing TCP connections. In [4] it was observed that the utilization of a link improves very fast with the increase of the buffer size until a certain threshold value. After that threshold value the further increase of the buffer size does not improve the link utilization but increases the queueing delay. It was also observed in [4] that one must ensure a certain minimum of space in the buffer for short connections and any further increase in the buffer capacity does not really improve the performance of short connections. Furthermore, two contradictory guidelines for the choice of the buffer size have been proposed. In [11] a connection-proportional buffer size allocation is proposed, whereas in [3] it was suggested that the buffer size should be set to the BDP of the outgoing link divided by the square root of the number of TCP connections. A rationale for the former recommendation is that in order to avoid a high loss rate the buffer must accommodate at least few packets from each connection. And a rationale for the latter recommendation is based on the reduction of the synchronization of TCP connections when the number of connections increases. Then, [11], [3] were followed by two works [6], [8] which try to reconcile these two contradictory approaches. In particular, the authors of [6] recommend to follow the rule of [3] for a relatively small number of long-lived connections and when a number of long-lived bottlenecked connections is large, to switch to the connection-proportional allocation. One of the main conclusions of [8] is that there are no clear criteria for the optimization of the buffer size. Then, the author of [8] proposed a general avenue for research on the router buffer sizing: "Find the link buffer size that accommodates both TCP and UDP traffic." We note that UDP (User Datagram Protocol) [14] does not use any congestion control and reliable retransmission and it is mostly employed for delay sensitive applications such as Internet Telephony.

All the above mentioned works on the router buffer sizing are based on quite rough approximations and do not strictly speaking take into account the feedback nature of TCP pro-

0-7803-9568-9/05/$20.00 ©2005 IEEE

tocol. Here we propose a mathematically solid framework to analyze the interaction of TCP with the finite buffer of an IP router. In particular, we state a criterion for the choice of the optimal buffer size in a mathematical form. Our optimization criterion can be considered as a mathematical formalization of the lingual criterion proposed in [8]. Furthermore, the Pareto set obtained for our model allows us to dimension the IP router buffer size to accommodate both data traffic and real time traffic.

The rest of the paper is organized as follows: In Section II, we state and solve a mathematical model of the interaction between TCP and the router buffer with a finite size. In particular, we show how an optimal buffer size can be chosen. Section III confirms the theoretical results of Section II with numerical examples and NS simulations. We conclude the paper with Section IV.

II. MATHEMATICAL MODEL

Let n long-lived TCP connections share a bottlenecked Internet router with the buffer size B and the transmission capacity μ. Denote by $\lambda_i(t)$ the instantaneous sending rate of connection $i = 1, ..., n$ at time $t \in [0, \infty)$. We consider a fluid model. Namely, data is represented by a fluid that flows into the buffer with the rate $\lambda(t) = \sum_{i=1}^{n} \lambda_i(t)$, and it leaves the buffer with the constant rate μ, if there is a backlog in the buffer. It is shown in [2] that the fluid model adequately describes the evolution of the TCP sending rate if the average sending rate is large enough. Denote by $x(t)$ the amount of data in the buffer at time $t \in [0, \infty)$. Then, the evolution of x is described by the following differential equation

$$\dot{x} = \begin{cases} \lambda - \mu, & \text{if } x > 0, \text{or if } x = 0 \text{ and } \lambda > \mu, \\ 0, & \text{if } x = 0 \text{ and } \lambda \leq \mu. \end{cases} \quad (1)$$

If $x < B$, the sending rate of connection i increases linearly in time with rate α_i. The constant α_i can be expressed in terms of the Round Trip Time (RTT) of the corresponding TCP connection [2]. Namely, $\alpha_i = 1/(RTT_i)^2$, where RTT_i is the Round Trip Time of connection i. Thus, if $x < B$,

$$\dot{\lambda} = \alpha, \quad (2)$$

where $\alpha = \sum_{i=1}^{n} \alpha_i$. When x reaches B, a congestion signal is sent to one or several TCP connection. Upon the reception of the congestion signal at time t, TCP connection reduces its sending rate by a multiplicative factor $\beta_0 \in (0, 1)$, that is, $\lambda_i(t + 0) = \beta_0 \lambda_i(t - 0)$. In the current TCP implementation $\beta_0 = 0.5$ [1]. Dynamical systems that combine both discrete and continuous behavior are known as Hybrid Systems [16]. Let us assume that when $x = B$ congestion signals are sent to $\tilde{n} \in \{1, ..., n\}$ connections and the sending rates of connections are distributed uniformly at the congestion moment. Then, the total sending rate is reduced on average by the factor

$$\beta = 1 - (1 - \beta_0)\frac{\tilde{n}}{n}.$$

And since in the fluid model all variables stand for average values, we can write that $\lambda(t + 0) = \beta\lambda(t - 0)$, when t is a moment of congestion.

Let us now formulate a performance criterion. On one hand, we are interested to obtain as large throughput as possible. That is, we are interested to maximize the average sending rate

$$\bar{\lambda} = \lim_{t\to\infty} \frac{1}{t} \int_0^t \lambda(s)ds.$$

On the other hand, we are interested to make the delay of data in the buffer as small as possible. That is, we are also interested to minimize the average amount of data in the buffer

$$\bar{x} = \lim_{t\to\infty} \frac{1}{t} \int_0^t x(s)ds.$$

Clearly, these two goals are contradictory. In fact, here we face a typical example of multicriteria optimization. A standard approach to multicriteria optimization is to consider the optimization of one criterion under constraints for the other criteria (see e.g., [13]). Namely, we would like to maximize the throughput given that the average amount of data in the buffer does not exceed a certain value

$$\max\{\bar{\lambda} : \bar{x} \leq \bar{x}_*\}. \quad (3)$$

Or we would like to minimize the average delay given that the average throughput is not less than a certain value

$$\min\{\bar{x} : \bar{\lambda} \geq \bar{\lambda}_*\}. \quad (4)$$

The solution to the above constrained optimization problems can be obtained from the Pareto set of the following unconstrained optimization problem

$$\max\left\{\lim_{t\to\infty} \frac{1}{t} \int_0^t c_1\lambda(s) - c_2 x(s)ds\right\}. \quad (5)$$

All three optimization problems (3), (4) and (5) can be regarded as mathematical formulation of the lingual criterion "find the link buffer size that accommodates both TCP and UDP traffic" given in [8]. Since UDP traffic does not contribute much in terms of the load, for the design of IP routers one can use for instance optimization problem (3) where the delay constraint is imposed by the UDP traffic.

We note that here we deal with the optimal impulse control problem of a deterministic system with long-run average optimality criterion. To the best of our knowledge there are no available results on such type of problems in the literature. In principle, the control policy in our model can depend on x and λ. In practice, however, all currently implemented buffer management schemes (e.g., AQM, DropTail) send congestion signals based only on the state of the buffer. Thus, we also limit ourselves to the case when the control depends only on the amount of data in the buffer.

In the case of the DropTail buffer management policy, it is possible to express the average sending rate and the average buffer load as functions of the buffer size. These functions are given in the next theorem.

Theorem 1: Let B be the router buffer size and the DropTail buffer management policy is used in the router.

Then, the average sending rate and the average amount of data in the buffer are given by

$$\bar{\lambda} = \mu, \tag{6}$$

$$\bar{x} = B - \frac{2}{3}B_0, \tag{7}$$

when $B \in [B_0, \infty)$, where $B_0 = \frac{\mu^2}{2\alpha}\frac{(1-\beta)^2}{(1+\beta)^2}$, and $\bar{\lambda}$ and $\bar{x}$ are given by

$$\bar{\lambda} = \frac{1+\beta}{2}(\alpha\theta + \mu), \tag{8}$$

$$\bar{x} = \frac{\alpha}{6(\tau+\theta)}[\theta^3 + \tau\theta^2 + 2(\tau - \sqrt{\tau^2-\theta^2})(\theta^2-\tau^2)], \tag{9}$$

when $B \in [0, B_0)$, where $\tau = (\mu - \mu\beta - \alpha\beta\theta)/\alpha$ and $\theta = \sqrt{2B/\alpha}$.

Proof: The fluid model (1) and (2) with the impulsive control of sending rate λ based solely on the buffer content x can evolve in three distinct periodic regimes. In the first regime, the size of buffer B, which determines the moments of the impulse control applications, is sufficiently large and the buffer is never empty (see Figure 1). In the second regime, the buffer becomes empty only at isolated points of time (see Figure 2). The second regime can be considered as a limiting case of the first regime. Then, in the third regime, the buffer stays empty during a non-zero length time interval of each cycle (see Figure 3).

One can show that no other cycles can take place. Given any initial conditions, it is clear that after some time $\lambda(t)$ becomes less than μ. Let $\hat{\lambda} < \mu$ be the value of λ immediately after a jump: $\hat{\lambda} = \lambda(t+0) = \beta\lambda(t-0)$ and calculate $\Phi(\hat{\lambda})$, the value immediately after the next jump. If the trajectory between the jumps does not touch the axis $x = 0$ then $\Phi(\hat{\lambda}) = \beta 2\mu - \beta\hat{\lambda}$. Since this map is contracting, there exists only one stable point λ^* (denoted below as $\lambda(0)$) which defines the cycle. If the trajectory does touch the horizontal axis, the reasoning must be slightly modified and the system converges to a cycle of the third type.

Let us study the first regime. Without loss of generality, we can consider only one cycle that starts at time 0 just after the sending rate reduction and finishes at the time moment T of the next rate reduction. Thus, we have that

$$x(0) = x(T) = B. \tag{10}$$

Between two consecutive rate reductions the system evolves according to the differential equations (1) and (2), and hence,

$$\begin{cases} \lambda(t) = \lambda(0) + \alpha t, \\ x(t) = x(0) + (\lambda(0) - \mu)t + \frac{\alpha}{2}t^2, \end{cases}$$

for $t \in [0, T)$. From condition (10) we determine that the duration of the cycle is $T = 2(\mu - \lambda(0))/\alpha$. Then, from the condition $\lambda(0) = \beta\lambda(T-0)$, we get

$$\lambda(0) = \frac{2\beta\mu}{1+\beta}$$

and, consequently, $T = \frac{2\mu}{\alpha}\frac{1-\beta}{1+\beta}$. Next, we calculate the minimal amount of data in the buffer B_{min}, which is achieved at the middle of the cycle.

$$B_{min} = x(T/2) = B - B_0 = B - \frac{\mu^2}{2\alpha}\frac{(1-\beta)^2}{(1+\beta)^2} \tag{11}$$

Since in this regime the buffer is never empty, $\bar{\lambda} = \mu$. Then, one can easily calculate the average amount of data in the buffer

$$\bar{x} = B + \frac{\lambda(0) - \mu}{2}T + \frac{\alpha}{6}T^2 = B - \frac{2}{3}B_0.$$

In particular, the equation (11) provides an expression for the value of B_0, which defines the limiting regime when the buffer becomes empty only at isolated points of time (see Figure 2).

Next, let us study the third regime. Points *A* and *D* in Figure 3 correspond to the beginning and the end of a cycle as defined for the first regime. At point *B* the buffer becomes empty, and at point *C* the sending rate again becomes equal to the transmission capacity μ and the amount of data in the buffer starts to grow from zero.

Denote by τ the time of the system transition from *A* to *C* and denote by θ the time of the system transition from *C* to *D*.

For two segments of the system trajectory *A*-*B* and *C*-*D*, we can write

$$\begin{cases} \lambda(\tau) = \beta\lambda(T) + \alpha\tau = \mu, \\ \lambda(T) = \mu + \alpha\theta. \end{cases}$$

Thus, we have that $\tau = (\mu - \beta\mu - \alpha\beta\theta)/\alpha$. Taking into account that $\lambda(0) = \beta(\mu + \alpha\theta)$ and $\tau + \theta = (1-\beta)(\theta + \mu/\alpha)$, one can calculate the average sending rate as a function of θ.

$$\lambda = \frac{1}{\tau+\theta}\int_0^{\tau+\theta}[\lambda(0) + \alpha s]\, ds = \frac{1+\beta}{2}(\alpha\theta + \mu)$$

Next, we note that

$$x(T) = \frac{1}{2}\alpha\theta^2 = B,$$

and, consequently, we can express θ as a function of B, that is, $\theta = \sqrt{2B/\alpha}$.

Let us now calculate the average amount of data in the buffer. Towards this end, denote by τ_0 the time of the system transition from *A* to *B*. The value of τ_0 is determined by the following equation

$$x(\tau_0) = x(0) + (\lambda(0) - \mu)\tau_0 + \frac{\alpha}{2}\tau_0^2$$
$$= \frac{\alpha}{2}\theta^2 - \alpha\tau\tau_0 + \frac{\alpha}{2}\tau_0^2 = 0.$$

Hence, $\tau_0 = \tau - \sqrt{\tau^2 - \theta^2}$. And then, we have

$$\bar{x} = \frac{1}{\tau+\theta}\left[\int_0^{\tau_0} x(s)ds + \int_\tau^T x(s)ds\right]$$
$$= \frac{\alpha}{6(\tau+\theta)}[\theta^3 + \tau\theta^2 + 2\tau_0(\theta^2 - \tau^2)].$$

This completes the proof. ■

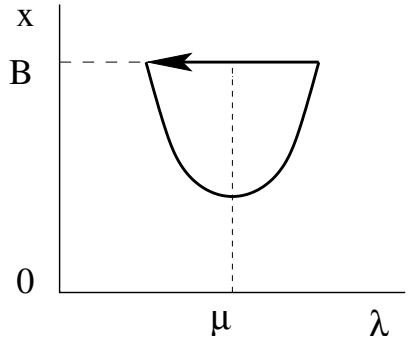

Fig. 1. The first regime ($B > B_0$).

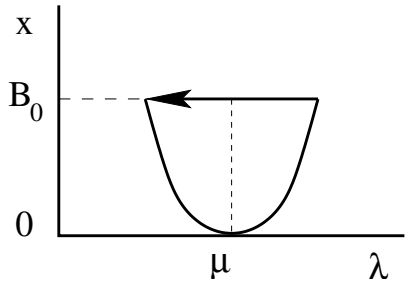

Fig. 2. The second regime ($B = B_0$).

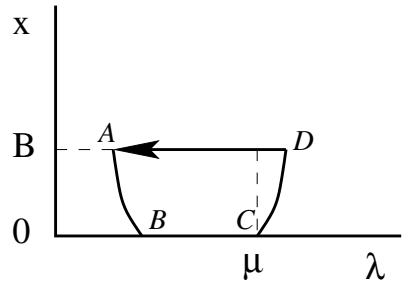

Fig. 3. The third regime ($B < B_0$).

Two limiting cases are of particular interest for us: (a) when the throughput is close to the router transmission capacity and (b) when the average amount of data in the buffer is close to zero. Let us first consider case (b).

Corollary 1: When $B \downarrow 0$, $\bar{x} \to 0$ and $\bar{\lambda} \to \frac{1+\beta}{2}\mu$.

Proof: The proof immediately follows from formulae (8) and (9). ■

Corollary 1 implies that even in the case of the zero buffer size and completely synchronized connections ($\tilde{n} = n$), the system utilization is quite high (around 75%, if $\beta_0 = 0.5$). If there are for instance five non-synchronized connections, the system utilization improves to 95% but the buffer stays empty. We recall that here we use a fluid model. In practice, the granularity of the data flow imposes some minimal constraints on the size of the buffer. Nevertheless, as was observed in [4], the utilization of a link improves very fast when the buffer size is increased from zero to some small value.

Corollary 2: When $B \uparrow B_0 = \frac{\mu^2}{2\alpha}\frac{(1-\beta)^2}{(1+\beta)^2}$, $\bar{x} \to B_0/3$ and $\bar{\lambda} \to \mu$.

We note that B_0 corresponds to the minimal size of the buffer when the link is fully utilized. If one cares only about the throughput of a TCP connection, then B_0 corresponds to the optimal choice of the buffer size. Let us study how the value of B_0 depends on the number of competing TCP connections.

First, let us consider the case of a single TCP connection when $\alpha = 1/RTT^2$ and $\beta = \beta_0 = 0.5$. In this case, we have $B_0 = (\mu RTT)^2/18$.

Next, let us study how much buffer space one needs to allocate for multiple TCP connection. To get a clear dependence of the optimal buffer size on the number of TCP connections, we Analise the symmetric case when all TCP connections have the same Round Trip Times. Thus, we have that $\alpha = n/RTT^2$. If TCP connections are always synchronized, we have $\beta = 0.5$ and

$$B_0 = \frac{(\mu RTT)^2}{18n},$$

where n is the number of TCP connections. And if TCP connections are not synchronized, we have $\beta = 1 - 1/(2n)$ and

$$B_0 = \frac{(\mu RTT)^2}{2n(4n-1)^2} \sim \frac{(\mu RTT)^2}{32n^3} \quad \text{as} \quad n \to \infty. \qquad (12)$$

This is a surprising result, as in [3] the asymptotics for the optimal buffer size is inversely proportional to $\sqrt{n}$. Thus, the present model recommends that one can choose even smaller buffer sizes than suggested in [3]. Of course, this recommendation is true as long as the fluid model is valid. Similar observations on the applicability of the rule of [3] were made in [6] and in [8].

Another interesting fact is that the asymptotics $B_0 \sim 1/\sqrt{n}$ suggested in [3] is explained by the desynchronization effect. From the present model it is also clear that the desynchronization helps a lot. However, the present model also implies that even if TCP connections are completely synchronized, the optimal buffer size decreases as $1/n$ when n, the number of TCP connections, increases. It is stated in [3] that in the case of synchronized connections one should follow the BDP rule as for the case of a single connection. The present model as well as the simulations of the ensuing Section III do not appear to confirm this statement.

Changing B from 0 up to B_0, we can plot the Pareto set, which has a shape as in Figure 4.

Next we illustrate that the difference in the value of B_0 is a direct consequence of the traffic model used to characterize the dynamics of the TCP sending rate and the interaction of TCP with packet losses.

The BDP rule of thumb [17] can be explained with the help of the following model. Consider a network with a single TCP connection. As above, let $\lambda(\tau)$, μ and RTT denote

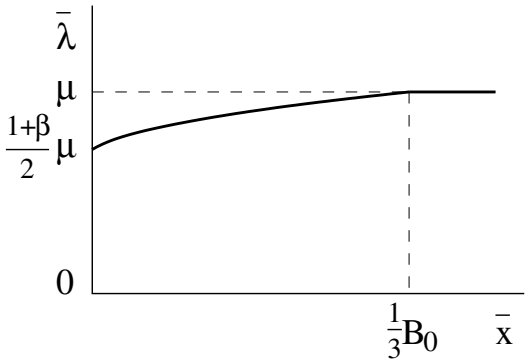

Fig. 4. The Pareto set.

the sending rate at time τ, the capacity of the link and the round trip time, respectively. Based on the behavior of the TCP protocol, roughly speaking, upon experiencing a packet loss a TCP sender stops sending data during a time period equal to RTT. After this inactivity period, the sender resumes transmitting at a rate approximately equal to the capacity of the link (see Figure 5). Hence, in order the link to be fully utilized during the amount of time in which the sender is not transmitting, the size of the buffer must be equal to the dashed area in Figure 5. Namely, we have

$$B_0 \geq RTT \times \mu,$$

which is commonly referred as the bandwidth-delay product rule of thumb. We refer to [17, Section 3] and [3, Section 1.2] for a more detailed discussion on the derivation of the rule-of-thumb.

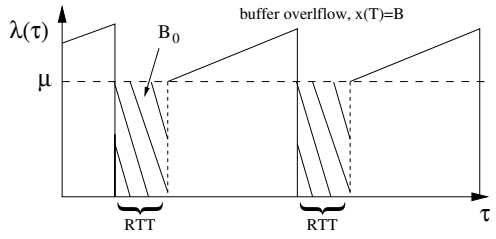

Fig. 5. Representation of B_0.

In our model, the evolution of the sending rate and the queue length are governed by equations (1) and (2). In Figure 6 we represent the dynamics of the sending rate in time.

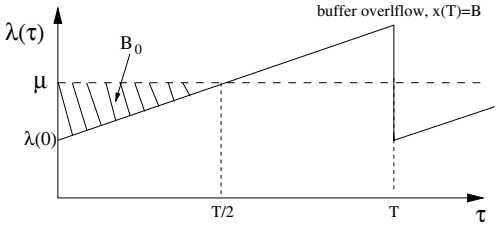

Fig. 6. Representation of B_0.

Then, similarly to the previous case, the value of B_0 corresponds to the dashed area. And consequently, we have

$$\begin{aligned} B_0 &= \mu T/2 - \int_0^{T/2} \lambda(s)ds \\ &= \mu T/2 - \lambda(0)T/2 - \alpha/2(T/2)^2 \\ &= T^2\alpha/8 = \frac{\mu^2}{2\alpha}\frac{(1-\beta)^2}{(1+\beta)^2}, \end{aligned}$$

which is precisely the value obtained in equation (11).

Of course, each model has its own limitations. We think that the first model is more appropriate in the case of a single TCP connection, whereas our model is more suitable in the case of multiple TCP connections. This statement is confirmed by the simulations presented in the next section.

III. SIMULATIONS

We perform network simulations with the help of NS-2, the widely used open-source network simulator [12]. We consider the following benchmark example of a TCP/IP network with a single bottleneck link. The topology may for instance represent an access network. The capacity of the bottleneck link is denoted by μ and its propagation delay is denoted by d. The capacities of N links leading to the bottleneck link are supposed to be large enough (or the load on each access link is small enough) so that they do not hinder the traffic. Each of these N links has a propagation delay d_i. We assume that in each access link there is one persistent TCP connection.

In the NS simulations we use the following values for the network parameters: bottleneck capacity is $\mu = 100$Mbps, bottleneck link propagation delay $d = 1$ms, the access link capacity and delay are 100Mbps and 1ms, respectively. The packet size is 500bytes and we use the New Reno flavor of TCP. The number of access links is equal to the number of connections. The fact that the delays in the access links are the same implies that the TCP connections will be synchronized.

In Figure 7 we depict the Pareto set for the cases of $N = 5$ and $N = 20$ connections. The qualitative shape of the curves agrees with what our model predicts. When the buffer size is 0, the achieved average sending rates are respectively 64.4Mbps and 66.6Mbps, slightly lower than 75Mbps, the value obtained in Corollary 1.

In our numerical example, let B_0 be the minimum buffer size that guarantees a link utilization greater than 99.9%. According to this definition, in our simulation the values of B_0 are equal to 120 ($N = 5$) and 70 ($N = 20$), while equation (12) gives a value of 250 ($N = 5$) and 62 ($N = 20$). We note that in this example the bandwidth delay product suggests that the buffer length should be set to 150 independently of the number of TCP connections. The "Stanford rule" provided in [3] ($\mu \times RTT/\sqrt{N}$) indicates that the buffer size should be set to 67 ($N = 5$) and 21 ($N = 20$), respectively.

The differences between the results obtained with the analytical model, and those obtained by simulations can be explained by the fact that the aggregated traffic in the

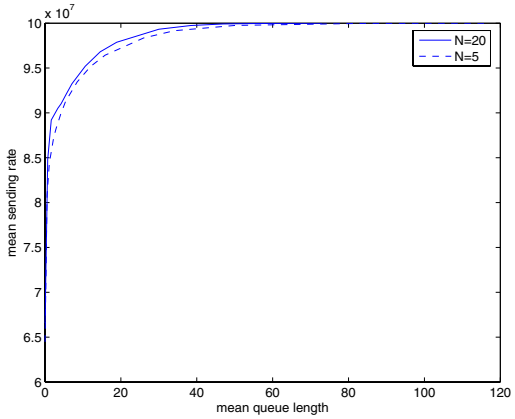

Fig. 7. The Pareto set for $N = 5$ and $N = 20$.

simulations is not as smooth as the fluid model we used in the model. Hence, when the buffer length is 0, the obtained average sending rate is smaller than the one obtained with the fluid model. Similarly, in the simulated scenario the minimum buffer length that guarantees full utilization of the link is larger than the one the fluid model predicts.

In Figure 8 we depict the relative error incurred by the three above mentioned methods. We note that the error of the fluid model reduces very quickly as the number of connections increases. In particular, Figure 8 confirms that the BDP rule is appropriate for the case of a single TCP connection but in the case of multiple TCP connections one should apply a different model.

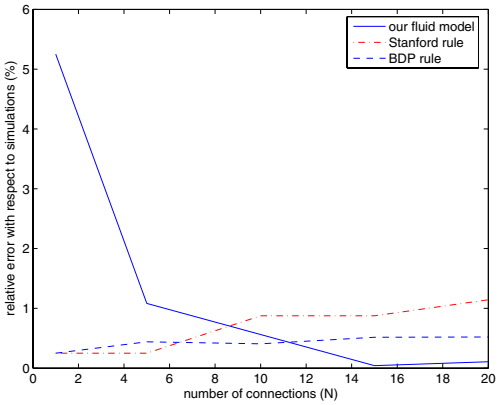

Fig. 8. Relative error of B_0

IV. CONCLUSIONS

In this paper we have formulated the problem of choosing the buffer size of routers in the Internet as a multi-criteria optimization problem. In agreement with previous works, our model suggests that as the number of long-lived TCP connections sharing the common link increases, the required minimum buffer size to achieve full link utilization reduces. We have shown that the various existing rule-of-thumbs in the literature are a direct consequence of the assumptions made to model the aggregate traffic that arrives into the bottleneck link. The simulations carried out confirm the qualitative insights drawn from our model. In particular, it seems that due to the fluid model approach used in our analysis, the obtained value of the minimal buffer size can be considered as a lower bound. The Pareto set obtained for our model allows us to dimension the IP router buffer size to accommodate real time traffic as well as data traffic.

REFERENCES

[1] M. Allman, V. Paxson and W. Stevens, TCP congestion control, *RFC 2581*, April 1999, available at `http://www.ietf.org/rfc/rfc2581.txt`.

[2] E. Altman, K.E. Avrachenkov and C. Barakat, "A stochastic model of TCP/IP with stationary random losses", in the Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, also in *Computer Communication Review*, v.30, no.4, pp.231-242, October 2000.

[3] G. Appenzeller, I. Keslassy and N. McKeown, "Sizing Router Buffers", ACM SIGCOMM '04, Portland, Oregon, September 2004. Also in Computer Communication Review, Vol. 34, No. 4, pp. 281-292, October 2004.

[4] K. Avrachenkov, U. Ayesta, E. Altman, P. Nain, and C. Barakat, "The effect of router buffer size on the TCP performance", in the Proceedings of LONIIS workshop, St. Petersburg, Jan. 29 - Feb. 1, 2002.

[5] M. Christiansen, K. Jeffay, D. Ott and F. Donelson Smith, "Tuning RED for Web Traffic", *IEEE ACM Transactions on Networking*, v.9, no.3, pp.249-264, June 2001. An earlier version appeared in Proc. of ACM Sigcomm 2000.

[6] A. Dhamdhere, H. Jiang, and C. Dovrolis, "Buffer Sizing for Congested Internet Links", in the Proceedings of IEEE Infocom, Miami FL, March 2005.

[7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, v.1, no.4, pp.397–413, 1993.

[8] S. Gorinsky, A. Kantawala, and J. Turner, "Link buffer sizing: a new look at the old problem", Technical Report WUCSE-2004-82, Department of Computer Science and Engineering, Washington University in St. Louis, December 2004.

[9] V. Jacobson, Congestion avoidance and control, *ACM SIGCOMM'88*, August 1988.

[10] M. May, J. Bolot, C. Diot and B. Lyles, "Reasons Not to Deploy RED", in Proceedings of 7th International Workshop on Quality of Service (IWQoS'99), June 1999, London, UK.

[11] R. Morris, "Scalable TCP congestion control", In Proceedings of IEEE INFOCOM 2000, March 2000, Tel-Aviv, Israel.

[12] "Network Simulator, Ver.2, (NS-2) Release 2.18a", Available at: http://www.isi.edu/nsnam/ns/index.html.

[13] A.B. Piunovskiy, *Optimal Control of Random Sequences in Problems with Constraints.* Kluwer Academic Publishers: Dordrecht, 1997.

[14] J. Postel, User Datagram Protocol, *RFC 768*, August 1980, available at `http://www.ietf.org/rfc/rfc0768.txt`.

[15] K. Ramakrishnan, S. Floyd and D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, *RFC 3168*, September 2001, available at `http://www.ietf.org/rfc/rfc3168.txt`.

[16] A. van der Schaft and H. Schumacher. *An Introduction to Hybrid Dynamical Systems.* Lecture Notes in Control and Information Sciences 251, Springer-Verlag, 2000.

[17] C. Villamizar and C. Song, "High Performance TCP in the ANSNET", *ACM SIGCOMM Computer Communication Review*, v.24, no.5, pp.45–60, November 1994.