

# The Effect of Router Buffer Size on HighSpeed TCP Performance

Dhiman Barman    Georgios Smaragdakis    Ibrahim Matta

Computer Science Department  
Boston University  
{dhiman, gsmaragd, matta}@cs.bu.edu

**Abstract**—We study the effect of the IP router buffer size on the throughput of HighSpeed TCP (HSTCP). We are motivated by the fact that in high speed routers, the buffer size is important as such a large buffer size might be a constraint. We first derive an analytical model for HighSpeed TCP and we show that for small buffer size equal to 10% of the bandwidth-delay product, HighSpeed TCP can achieve more than 90% of the bottleneck capacity. We also show that setting the buffer size equal to 20% can increase the utilization of HighSpeed TCP up to 98%. On the contrary, setting the buffer size to less than 10% of the bandwidth-delay product can decrease HighSpeed TCP’s throughput significantly. We also study the performance effects under both DropTail and RED AQM. Analytical results obtained using a fixed-point approach are compared to those obtained by simulation.

Keywords: Congestion Control; High-Speed Networks; Transmission Control Protocol (TCP); Throughput Analysis

## I. INTRODUCTION

With the advent and deployment of Gigabit links, the capacity grows by orders of magnitude. Such high-capacity links have been used to transfer huge volumes of data [1]. TCP increase rule used to probe for available bandwidth becomes inefficient at such link speeds. TCP takes thousands of round-trip time (RTT) to reach full link utilization. Therefore, studies on TCP-like protocol performance are important to understand scaling behavior from megabits to gigabits per second. Many proposals [2], [3], [4], [5] on TCP have been propounded for high-bandwidth networks. In this paper, we are interested primarily in the performance of HighSpeed TCP [2]. Specifically, we analyze the effect of the IP router buffer size on the performance of long-lived HighSpeed TCP connections.

In high bandwidth-delay product networks, due to unrealistic constraints of TCP’s response function, TCP cannot open its window large enough to utilize the available bandwidth. In order to address this fundamental problem of TCP and TCP’s response function, TCP is enhanced to HighSpeed TCP [2] which modifies the window dynamics for high congestion window (low loss) regime. After a certain window threshold, HighSpeed TCP increases its congestion window aggressively to grab the available bandwidth and on losses, it reduces the window slowly to remain smooth. Since HighSpeed TCP’s window dynamics come into effect only at large window (low loss) regime, HighSpeed TCP does not modify TCP behavior in environments with high to mild congestion (typical of low-speed networks.)

In high bandwidth-delay networks in which HighSpeed TCP sends bursts of large number of packets, the amount of buffer available in the bottleneck router is an important issue to

keep the router highly utilized during congestion periods. The current recommendations require router manufacturers to provide a buffer whose size is comparable to the bandwidth-delay product (BDP) of the path which scales linearly with line speeds. As the link capacity increases to tens of Gbps, providing such a huge buffer may drastically increase the cost of the routers and impose technological problems such as heat dissipation, on-chip board space and increased memory access latency. A large buffer increases delay and delay variance which adversely affects real-time applications (e.g., video games, device control and video over IP applications.) It is therefore important to investigate the effects of buffering on HighSpeed TCP performance such as throughput, convergence to fairness and interaction in the presence of RED. Earlier proposals primarily focused on improving the performance of HighSpeed TCP or other variants to scale TCP in high-speed environments, e.g., [5], [6], [7]. None of these studies explicitly examined the effect of buffer size on the proposed TCP variants. We derive an analytical model for HighSpeed TCP and use a fixed-point method to numerically solve for the utilization achieved by competing long-lived HSTCP flows—The model captures how the performance of HighSpeed TCP is affected by the buffer through RTT.

The rest of the paper is organized as follows. In Section II, we derive an analytical model to compute the steady-state throughput of a single HSTCP connection given a fixed packet loss probability. In Section III, we describe the fixed-point method to compute the utilization of  $N$  HSTCP flows competing for a bottleneck with a given buffer size and also present numerical and simulation results. In Section IV, we describe the fixed-point method to compute the utilization of  $N$  HSTCP flows competing for a bottleneck having RED AQM with given buffer size and also present the numerical and simulation results. We discuss future work and conclude the paper in Section V.

## II. PERSISTENT HIGH SPEED TCP (HSTCP) CONNECTIONS

In this section, we analyze long-lived persistent HighSpeed TCP connections. Let us consider  $N$  persistent HSTCPs that are traversing the same bottleneck link of capacity  $C$ . Denote by  $R_i, i = 1, \dots, N$ , the average round-trip time of the  $i$ -th connection, and by  $p_i$ , the packet loss probabilities. The goal is to obtain the average sending rate of a persistent HSTCP from its window dynamics. In this section, we first analyze the throughput of a single HSTCP connection, then in Sections III and IV we consider the performance of  $N$  competing HSTCPs under Drop-Tail and RED routers, respectively.

### A. Analytical Model

In HighSpeed TCP, when the current congestion window size is less than  $w_{low}$ , the window size changes according to the same algorithm as that of TCP Reno. On the other hand, when the current congestion window is larger than  $w_{low}$ , it increases its congestion window more quickly, and decreases it more slowly than TCP Reno. Therefore, HSTCP is expected to achieve higher throughput than TCP Reno by keeping the congestion window size at a larger value. The degree of increase/decrease of the congestion window size depends on its current value. That is, when the congestion window size is larger, HighSpeed TCP increases it more quickly, and decreases it more slowly. Denote by  $p_{low}$  and  $p_{high}$ , the maximum loss probability at  $w_{low}$  and  $w_{high}$ , respectively. In [2], the increase amount of the congestion window in each RTT is defined as  $a(w)$ , and the decrease amount of the congestion window as  $b(w)$ , when its current congestion window is  $w$ . That is, HighSpeed TCP increases its window size  $w$  by  $a(w)$  packets in one RTT when no packet loss is detected and decreases it to  $(1-b(w))w$  when it detects packet loss by duplicate ACK packets. Contrary to regular/standard TCP, where  $a(w) = 1$  and  $b(w) = 0.5$  regardless of the value of the congestion window  $w$ , HighSpeed TCP uses these same standard TCP values for  $a(w)$  and  $b(w)$  only for congestion window  $w \leq w_{low}$ .

We define  $w_t$  as the window size at time  $t$  and as  $R$  the round-trip time of the flow. The congestion window increases as follows [2]:

$$w_{t+R} = \begin{cases} 2w_t & \text{if } w_t < ssth_{HS} \\ w_t + 1 & \text{if } w_t \geq ssth_{HS} \text{ and } w_t \leq w_{low} \\ w_t + a(w_t) & \text{if } w_t \geq ssth_{HS} \text{ and } w_t > w_{low} \end{cases} \quad (1)$$

The first condition denotes the Slow Start Phase and the two following conditions the Congestion Avoidance Phase.  $ssth_{HS}$  is the ssthresh of HighSpeed TCP. Moreover,  $a(w)$  and  $b(w)$  are given as follows:

$$a(w) = \frac{2w^2b(w)p(w)}{2-b(w)}; b(w) = \frac{\log(\frac{w}{w_{low}})}{\log(\frac{w_{high}}{w_{low}})}(b_{high} - 0.5) + 0.5$$

$$p(w) = \exp\left(\frac{\log(\frac{w}{w_{low}})}{\log(\frac{w_{high}}{w_{low}})} \log\left(\frac{p_{high}}{p_{low}}\right) + \log(p_{low})\right)$$

$$p_{low} = \frac{1.5}{w_{low}^2}$$

$p_{high}$ ,  $w_{high}$ ,  $b_{high}$  and  $w_{low}$  are the parameters of HighSpeed TCP such that the congestion window size of  $w_{high}$  packets is always achieved with packet error probability  $p_{high}$ . Upon a packet loss event, HighSpeed TCP decreases its congestion window as follows:

$$w_{t+\delta t} = (1-b(w_t))w_t$$

Next, we analyze the throughput of a single HighSpeed TCP connection as a function of the loss rate it experiences. Our

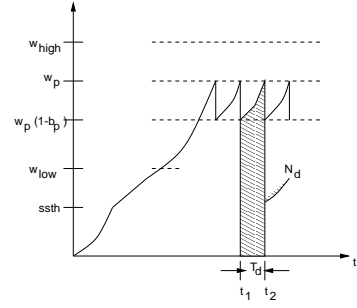


Fig. 1. Window dynamics of HighSpeed TCP

approach is similar to the one used in [8]. For simplicity, we substitute  $w_h = w_{high}$ ,  $w_l = w_{low}$ ,  $A = \log(w_h/w_l)$ ,  $y = \frac{\log(\frac{w}{w_l})}{\log(\frac{w_h}{w_l})}$ ,  $b_h = b_{high}$ ,  $c = \log(\frac{p_{high}}{p_{low}})$ , and  $k = \log(p_{low})$ . Sometimes we will denote  $w(p)$  as  $w_p$  and  $b(p)$  as  $b_p$ . The first two conditions of Equation (1) are the same as with TCP Reno and can be analyzed as in [9]. For our purposes we focus on the steady-state behavior beyond  $w_l$ . Using the rules of HighSpeed TCP in Equation (1), we obtain a continuous fluid approximation and linear interpolation of the window (Figure 1) between  $w_t$  and  $w_{t+R}$ :

$$\frac{dw(t)}{dt} = \frac{a(w(t))}{R} = \frac{2w^2b(w)p(w)}{R(2-b(w))}$$

From the substitutions given above, we find that:

$$b(w) = y(b_h - 0.5) + 0.5; \quad p(w) = \exp(y(c + k))$$

$$w = w_l e^{yA}; \quad \frac{dw}{dy} = w_l e^{yA} A \frac{dy}{dt}$$

$$\frac{w_l e^{yA} A dy}{dt} = \frac{2(w_l e^{yA})^2 (y(b - \frac{1}{2}) + \frac{1}{2}) e^{(y(c+k))}}{R(\frac{3}{2} - y(b - \frac{1}{2}))}$$

$$\frac{(3/2 - y(b - 1/2)) e^{-y(A+c)}}{y(b - 1/2) + 1/2} dy = 2 \frac{w_l p_l dt}{AR} \quad (2)$$

We find an upper bound as follows:

$$\frac{3 - (2b - 1)y}{1 + (2b - 1)y} e^{-y(A+c)} \leq \frac{2 - b}{b} e^{-y(A+c)}$$

because  $y \in [0,1]$  and typical value of  $b$  is proposed to be  $b = 0.1 \leq 0.5$  [2]. Using this approximation we can integrate Equation (2):

$$-\frac{1}{(A+c)} e^{-y(A+c)} = \frac{2bw_l p_l t}{(2-b)AR} + \mu$$

where the interpolation constant  $\mu$  is equal to  $\frac{-1}{A+c} e^{-(1+c/A) \log(w_p(1-b_p)/w_l)}$ , for  $t = 0$ . Since we are interested in the steady state of the window dynamics, we assume that a loss epoch starts at  $t = 0$  ( $t_1$  in Figure 1 without time-shifting the curve along the horizontal axis) with window value of  $w_p(1-b(w_p))$  where  $w_p$  is the maximum value of the window  $w_t$  at time  $t_2$  as shown in Figure 1, if

the loss rate is  $p$ . Using the previous equation we can derive  $t$  as function of  $y$  and vice versa:

$$t = \frac{(2-b)AR}{2(A+c)bw_l p_l} \left[ e^{-(1+c/A) \log(\frac{w_p(1-bp)}{w_l})} - e^{-y(A+c)} \right]$$

From the above equation, expressing  $y$  in terms of  $t$ , we get

$$\begin{aligned} e^{-y(A+c)} &= e^{\log(\frac{w_l}{w_l}) - \frac{A+c}{A} t} \\ \left(\frac{w(t)}{w_l}\right)^{-\frac{A+c}{A}} &= \left[ e^{-(1+c/A) \log(\frac{w_p(1-bp)}{w_l})} - \frac{2b(A+c)w_l p_l t}{AR(2-b)} \right] \\ w(t) &= w_l \left[ e^{-(1+c/A) \log(\frac{w_p(1-bp)}{w_l})} - \frac{2b(A+c)w_l p_l t}{AR(2-b)} \right]^{\frac{-A}{A+c}} \end{aligned}$$

For our analysis, we are particularly interested in the parameters  $T_d$  and  $N_d$  marked in Figure 1.  $T_d$  and  $N_d$  are the time and the number of packets between two successive packet drops, respectively. Both these parameters are independent of time-shifting the curve along the horizontal axis. Referring to Figure 1, we can arrange the curve such that at  $t_1 = 0$ ,  $w(t_1) = w_p(1-b(w_p))$ . Thus  $T_d = t_2 - t_1 = t_2$ , where  $t_2$  is given by:

$$t_2 = \frac{(2-b)AR}{2(A+c)bw_l p_l} \left[ e^{-(1+\frac{c}{A}) \log(\frac{w_p(1-bp)}{w_l})} - e^{-(1+\frac{c}{A}) \log(\frac{w_p}{w_l})} \right]$$

$N_d$  is the shaded area under the curve in Figure 1 which is given by the following integral:

$$N_d = \int_{t_1}^{t_2} \frac{w(t)}{R} dt$$

Replacing  $e^{-\frac{c}{A} \log(\frac{w_p(1-bp)}{w_l})}$  with  $\alpha$ ,  $e^{-\frac{c}{A} \log(\frac{w_p}{w_l})}$  with  $\beta$  and  $\frac{-A}{A+c}$  with  $\gamma$ , we get:

$$\begin{aligned} N_d &= \int_0^{t_2} w_l (\alpha - \beta t)^{-\gamma} dt = \frac{w_l [(\alpha - \beta t_2)^{-\gamma+1} - \alpha^{-\gamma+1}]}{R\beta(1-\gamma)} \\ &= \frac{w_l(A+c)}{R\beta c} \left[ e^{-\frac{c}{A} \log(\frac{w_p}{w_l})} - e^{-\frac{c}{A} \log(\frac{w_p(1-bp)}{w_l})} \right] \\ &= \frac{A(2-b)}{2bc p_l} \left[ e^{-\frac{c}{A} \log(\frac{w_p(1-bp)}{w_l})} - e^{-\frac{c}{A} \log(\frac{w_p}{w_l})} \right] \end{aligned}$$

The average throughput  $\lambda$  in packets per second is the number of packets sent in each congestion epoch divided by the duration between drops. So we can express  $\lambda$  as follows:

$$\begin{aligned} \lambda &= \frac{N_d}{T_d} \\ &= \frac{w_l}{R} \left( 1 + \frac{A}{c} \right) \left[ \frac{e^{-\frac{c}{A} \log(\frac{w_p(1-bp)}{w_l})} - e^{-\frac{c}{A} \log(\frac{w_p}{w_l})}}{e^{-(1+\frac{c}{A}) \log(\frac{w_p(1-bp)}{w_l})} - e^{-(1+\frac{c}{A}) \log(\frac{w_p}{w_l})}} \right] \end{aligned}$$

The above equation leads to:

$$\begin{aligned} \lambda &= \frac{w_l}{R} \left( 1 + \frac{A}{c} \right) \frac{\left[ \left( \frac{w_p(1-bp)}{w_l} \right)^{-c/A} - \left( \frac{w_p}{w_l} \right)^{-c/A} \right]}{\left[ \left( \frac{w_p(1-bp)}{w_l} \right)^{-(1+c/A)} - \left( \frac{w_p}{w_l} \right)^{-(1+c/A)} \right]} \\ &= \frac{w_l}{R} \left( 1 + \frac{A}{c} \right) \left[ \frac{(1-bp)^{-c/A} - 1}{(1-bp)^{-(1+c/A)} - 1} \right] \frac{w_p}{w_l} \end{aligned}$$

We use the previously made substitution for  $p(w)$  given by  $p(w) = \exp((c/A) \log(w/w_l) + \log p_l)$  which gives  $w_p/w_l = (p/p_l)^{A/c}$ . Thus, we get:

$$\begin{aligned} \lambda &= \frac{w_l}{R} \left( 1 + \frac{A}{c} \right) \left( \frac{p}{p_l} \right)^{\frac{A}{c}} \left[ \frac{(1-bp)^{-c/A} - 1}{(1-bp)^{\frac{-1}{(1+c/A)} - 1}} \right] \\ &\approx \frac{w_l}{R} \left( \frac{p}{p_l} \right)^{\frac{\log(w_h/w_l)}{\log(p_h/p_l)}} \end{aligned} \quad (3)$$

### III. FIXED POINT APPROXIMATION WITH DROP TAIL

In this section, we describe the fixed-point iterative method to evaluate HSTCP assuming an M/M/1/K model of the bottleneck queue [10]. The load on the bottleneck link is given by:

$$\rho = \frac{1}{C} \sum_{i=1}^N \lambda_i(p, RTT_i) \quad (4)$$

From classical M/M/1/K results:

$$p = \frac{\rho^K(1-\rho)}{1-\rho^{K+1}}; \quad R_i = 2d_i + \frac{MSS}{C} \left( \frac{\rho}{1-\rho} - \frac{K\rho^K}{1-\rho^K} \right)$$

where  $d_i$  is the propagation delay of the  $i$ -th connection and MSS the maximum segment size. So the load can be derived from Eq. (4):

$$\rho = \frac{1}{C} \sum_{i=1}^N \lambda_i \left( \frac{\rho^K(1-\rho)}{1-\rho^{K+1}}, 2d_i + \frac{MSS}{C} \left( \frac{\rho}{1-\rho} - \frac{K\rho^K}{1-\rho^K} \right) \right) \quad (5)$$

We observe the right hand side of Equation (5) is continuous in  $\rho \in [0, 1]$ . The throughput,  $\lambda$ , of HSTCP is continuous in  $p$  and  $p$  is continuous in  $\rho$ . Therefore, at least there exists a fixed point for Eq. (5). The value of the derivative of the iterative function depends on the buffer size and therefore, the uniqueness of the fixed point of the function depends on the buffer size.

#### A. Evaluation

We simulate the effect of buffer sizes on the utilization achieved by 10 long-lived HighSpeed TCP connections starting at different times. The topology is shown in Figure 2. The common link,  $R1$ - $R2$  is 1Gbps with 50ms delay. The RTTs of the connections are different ranging from 115.5ms to 124.5 ms and average RTT is 120ms. The buffer size at  $R1$ - $R2$  is varied as a fraction of 12500 1KB-packets (80% of BDP of the largest RTT.) The BDP of just  $R1$ - $R2$  is

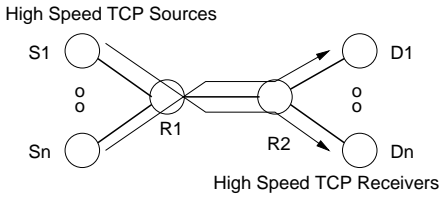


Fig. 2. Simulation Topology

12500 packets. In our experiments we use the values of  $w_h = 83000$ ,  $w_l = 38$ ,  $p_h = 10^{-7}$ , and  $p_l = 10^{-3}$ . Observing the bottleneck utilization as a function of buffer size in Figure 3, we find that results from simulations and our analytical model are quite close. In simulations, we find that the loss rate is mostly equal to  $p_l$ . We observe that buffer size equal to 10% of the bandwidth-delay product can sustain more than 90% utilization. The discrepancy between the two curves in

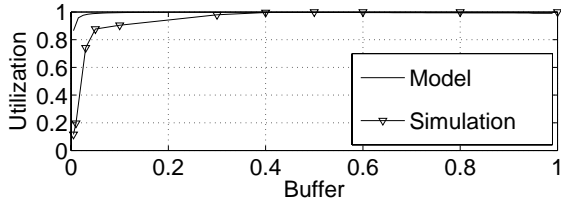


Fig. 3. Throughput as function of Buffer size fraction (max 12500)

Figure 3 mainly arises from our M/M/1/K assumption and simplifying approximations we made in deriving the closed-form throughput expression. In the next experiment, we change

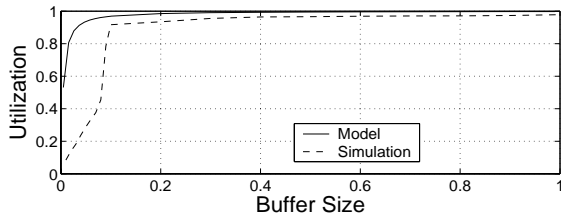


Fig. 4. Throughput as function of Buffer size fraction (max 6250)

the bandwidth of the common link  $R1-R2$  to 2.5Gbps and delay to 10ms. The RTTs of the connections are different ranging from 38ms to 142ms and average RTT is 90ms. The buffer size is varied as a fraction of 6250 1KB-packets (35% of BDP of largest RTT connection.) The BDP of just  $R1-R2$  is 6250 packets. We observe similar trends in the performance (see Figure 4) as in Figure 3.

We plot the window and queue evolution of two competing HighSpeed TCP flows in case of under-buffered (Figure 5) and over-buffered (Figure 6) bottleneck router using the topology shown in Figure 2. The RTT of HSTCP1 is almost twice that of HSTCP2. The bottleneck bandwidth is 1Gbps with 10ms of delay. The unfairness evident from the window dynamics is due to difference in RTTs of the two connections. We also see that a larger bottleneck buffer reduces the unfairness between

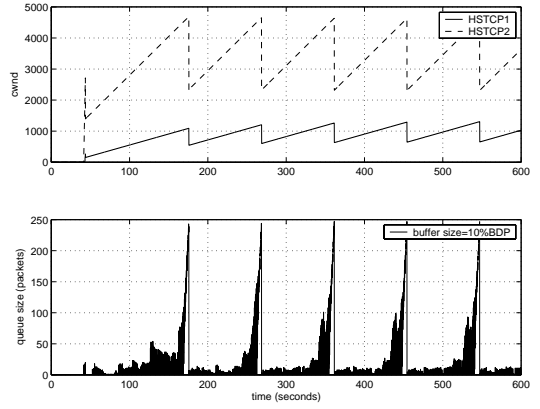


Fig. 5. Window behavior of two HighSpeed TCPs and droptail queue size in under-buffered case

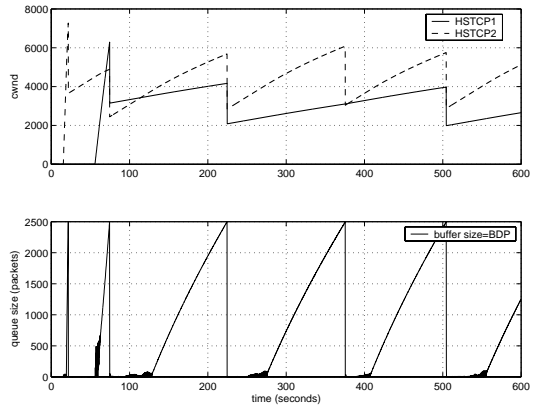


Fig. 6. Window behavior of two HighSpeed TCPs and droptail queue size in over-buffered case

two connections. Thus, one should consider the effect of buffer size on the tradeoff between RTT fairness and utilization.

#### IV. FIXED POINT APPROXIMATION WITH RED AQM

In this section, we evaluate the performance of HighSpeed TCP under RED. We follow the analytical model of RED in [11]. RED helps in removing the bias against bursty flows and burst sizes are large in the case of HighSpeed TCP flows. Using the PASTA property, the drop probability of a packet at a RED queue is given by:

$$p_{red} = \sum_{n=0}^K \pi(n)p(n)$$

$$p(n) = \begin{cases} 0 & \text{if } n \leq \min_{th} \\ \frac{(n - \min_{th})p_{max}}{\max_{th} - \min_{th}} & \text{otherwise} \end{cases}$$

In order to derive the utilization, we use a fixed-point iterative method:

$$\rho = \frac{1}{C} \sum_{i=1}^N \lambda_i(p_{red}, d_{red})$$

$$d_{red} = 2d_i + \frac{MSS}{C} \sum_{n=1}^K n\pi(n)$$

where  $\pi$  denotes the stationary distribution of the number of packets in the buffer.

### A. Evaluation

We simulate the effect of buffer size on the utilization achieved by 10 long-lived HighSpeed TCP connections under RED. The common link  $R1-R2$  is 1.0Gbps with 50ms delay. The buffer size is varied as a fraction of 12500 1KB-packets. The connections start at different times and also their RTTs are different ranging from 115.5ms to 124.5ms and average RTT is 120ms. We use Adaptive-RED in which  $\min_{th}$  and  $\max_{th}$  are dynamically adjusted. In our numerical evaluation we use simple RED for mathematical tractability; we set  $\min_{th} = 40\%$  of full buffer size and  $\max_{th}$  to full buffer size.

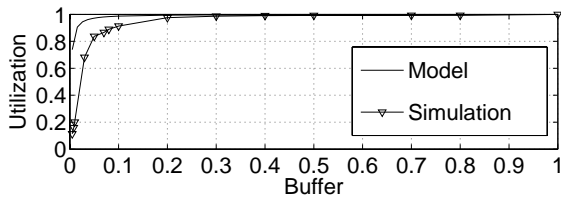


Fig. 7. Throughput as function of buffer size,  $\min_{th} = 0.4 \times \min(\text{BDP}, \text{buffer size})$ ,  $\max_{th} = \min(\text{BDP}, \text{buffer size})$ ,  $N = 10$ ,  $C = 1 \text{ Gbps}$

We plot the utilization of HighSpeed TCP in the presence of RED in Figure 7. The loss rate is  $10^{-5}$  for buffer ratio  $\geq 0.1$ .

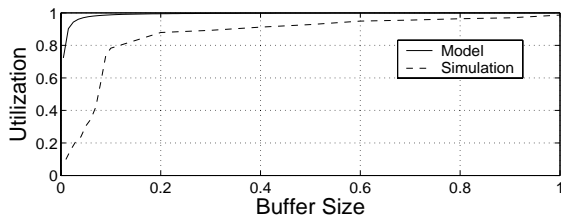


Fig. 8. Throughput as function of buffer size,  $\min_{th} = 0.4 \times \min(\text{BDP}, \text{buffer size})$ ,  $\max_{th} = \min(\text{BDP}, \text{buffer size})$ ,  $N = 10$ ,  $C = 2.5 \text{ Gbps}$

We plot a similar experiment in Figure 8 where we change the capacity to 2.5Gbps and delay to 10ms. The RTTs are different ranging from 38ms to 142ms and average RTT is 90ms. We vary the buffer size as a fraction of 6250 1KB-packets. We find that the loss rate is around  $10^{-6}$ . Despite using Adaptive-RED in the simulations, we observe close agreement with the numerical results.

### V. CONCLUSION

From our numerical results obtained using the fixed-point method as well as from simulations, we conclude that typically a persistent HighSpeed TCP connection crossing a bottleneck link has a low sending rate when the buffer size is small. The throughput increases as buffer size increases. We observe that a buffer size of 10% of the bandwidth-delay product can maintain at least 90% utilization. Increasing the buffer size beyond 10% increases the utilization marginally—reaching almost 98% when the buffer size is around 20% of the

bandwidth-delay product. On the other hand, a small buffer size below 10% of the bandwidth-delay product causes many packet drops, which lead to a drastic drop in the utilization and may reduce HighSpeed TCP to regular/standard TCP. We note that our results are consistent with a recent independent study [12], which shows that the rule-of-thumb of setting the buffer size equal to the bandwidth-delay product is not necessary.

We presented results considering a single bottleneck, however we believe that buffer requirements will have similar trends in the case of multiple congested bottlenecks but one needs to validate that. Although we only focused on long-lived flows, it will be interesting to investigate the performance of short-lived flows and constant-rate flows (e.g. UDP) on high-speed links with different buffer sizes. In our future work, we would also like to investigate the effects of buffer size on the performance of other recently proposed high-speed TCP variants, e.g. FAST TCP [4] which changes its window according to buffer delay.

### VI. ACKNOWLEDGEMENTS

This work was supported in part by NSF grants ANI-0095988, ANI-9986397, EIA-0202067, and ITR ANI-0205294. We are grateful to Lisong Xu and Injong Rhee for answering questions on the code used in [5].

### REFERENCES

- [1] *DataTAG project*, <http://www.datatag.org>.
- [2] S. Floyd, "HighSpeed TCP for Large Congestion Windows," in *RFC 3649, Experimental*, December 2003.
- [3] D. Katabi, M. Handley, and C. Rohrs, "Internet Congestion Control for Future High Bandwidth-Delay Product Environments," in *Proceedings of ACM SIGCOMM'02*, August 2002.
- [4] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," in *Proceedings of IEEE INFOCOM'04*, March 2004.
- [5] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Control for Fast, Long Distance Networks," in *Proceedings of IEEE INFOCOM'04*, 2004.
- [6] K. Tokuda, G. Hasegawa, and M. Murata, "Performance Analysis of HighSpeed TCP and its Improvement for High Throughput and Fairness against TCP Reno Connections," in *Proceedings of at High Speed Network Workshop (HSN 2003)*, November 2003.
- [7] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks," in *ACM SIGCOMM Computer Communication Review, Volume 33, Issue 2*, April 2003.
- [8] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms," in *Proceedings of IEEE INFOCOM'01*, March 2001, pp. 631–640.
- [9] T. V. Lakshman and U. Madhoo, "The Performance of TCP/IP for Networks with High Bandwidth-delay Products and Random Loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, July 1997.
- [10] K. Avrachenkov, U. Ayesta, E. Altman, P. Nain, and C. Barakat, "The Effect of Router Buffer Size on the TCP Performance," in *Proceedings of LONIS workshop on Telecommunication Networks and Teletraffic Theory, St.Petersburg, Russia*, pp.116-121, January 2002.
- [11] T. Bonald, M. May, and J.-C. Bolot, "Analytic Evaluation of RED Performance," in *Proceedings of IEEE INFOCOM'00*, 2000.
- [12] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proceedings of ACM SIGCOMM'04*, 2004.