Analysis of Scalable TCP in the presence of Markovian Losses

E. Altman K. E. Avrachenkov A. A. Kherani
B.J. Prabhu
INRIA Sophia Antipolis
06902 Sophia Antipolis, France.
Email:{altman,k.avratchenkov,alam,bprabhu}@sophia.inria.fr

Abstract

In high speed networks, the standard TCP's AIMD algorithm was observed to be inefficient in utilizing the link capacity. As a result of which several proposals such as High-Speed TCP, FAST and Scalable TCP, were put forward. In contrast to the additive increase multiplicative decrease algorithm used in the standard TCP, Scalable TCP uses a multiplicative increase multiplicative decrease (MIMD) algorithm for the window size evolution. In this paper, we first present an approximate expression for the throughput of a long lasting Scalable TCP session when the losses are i.i.d. and due to window dependent errors. We then present an analysis when the losses are due to Markovian window independent errors. We compare our analytical results with ns-2 simulations.

1 Introduction

The additive increase algorithm of the standard TCP has been observed to recover slowly from packet losses due to events not related to congestion. This phenomenon becomes pronounced when a TCP session transfers large files over high speed networks in which packet losses could occur due to link layer errors [1]. Since the losses are not due to congestion, the additive increase in the recovery phase after a window reduction results in inefficient use of the large bandwidth. In order to improve the utilization of the available capacity in high speed networks, modifications to the standard TCP have been proposed in [1]-[3]. In [4], Kelly has proposed a variation of TCP, called Scalable TCP, wherein upon each ACK it receives, the sender increases its congestion window (*cwnd*) by 0.01 packets. When a loss event is detected, the sender decreases *cwnd* by a factor of 0.125. Hence, if the window size is $W(t)$ at some time t (meaning that there are $W(t)$ unacknowledged packets in the network) then, in the absence of losses, the window size after an RTT (round-trip time), $W(t+RTT)$, would be $1.01 \times W(t)$, whereas if there are losses during $(t, t + RTT)$, $W(t + RTT)$ will be around $0.875 \times W(t)$ (here we assume that, as in New Reno and SACK, the window is reduced only once during a round trip time even if there are several losses). A feature of this algorithm is that, starting from a window size of some fraction of the bandwidth-delay product (BDP), the number of RTT*s* required to reach BDP is independent of the link speed.

The outline of this paper is as follows. In Section 2, we present the system model and some preliminary analysis. In Section 3, we analyse the performance of Scalable TCP in the presence of packet errors. These packet losses can be either due to link errors or some window based dropping/marking scheme. We provide an approximate expression of the throughput and compare it with *ns-2* simulations. In Section 4, we consider the loss process to be Markovian but window independent, and present a corresponding model. The simulation results are presented in Section 5.

2 System Model

We consider the scenario where a single FTP application transfers data using Scalable TCP. Scalable TCP uses Multiplicative Increase and Multiplicative Decrease(MIMD) congestion control in contrast to the AIMD algorithm of the standard TCP. Let W_n denote the sender's congestion window at the end of the n^{th} RTT. The recursive equation for W_n is given by

$$\begin{aligned} W_{n+1} &\leftarrow \alpha * W_n \quad \text{no losses,} \\ W_{n+1} &\leftarrow \beta * W_n \quad \text{one or more losses,} \end{aligned}$$

where $\alpha > 1$, and $\beta < 1$. Here we assume that the sender decreases the window only once in a RTT even if there were multiple losses. In [4], the author suggests using $\alpha = 1.01$ and $\beta = 0.875$.

Let A_n be a random variable such that $A_n = \alpha$ if there were no losses in the n^{th} RTT, and $A_n = \beta$ otherwise. We can rewrite the recursive equation as follows.

$$W_{n+1} = \min(A_n W_n, B_u), \tag{1}$$

where B_u is the upper bound on the sender's window. This upper bound may be due to the receiver's buffer limitation. In [5], the authors presented an analysis of Equation 1 when the the process A_n was independent of W_n and was i.i.d. In this paper, we extend the analysis to window independent Markovian losses and also to random window dependent losses. Random window dependent losses include losses due to packet errors which may occur on fibre optic links.

3 Window Dependent Random Losses : An Approximation

Our analysis is based on the results obtained in [5]. We consider that the losses are window dependent. Specifically, we assume that each packet is dropped (or, equivalently, is in error) with a constant probability q. As a consequence of this assumption, the probability of packet drops in an RTT are no longer independent of the window size in that RTT. Then, we propose an approximation to this model, which will enable us to compute the throughput in the window dependent model using the expression of throughput in the window independent model obtained in [5].

Let W_n be the window size in the n^{th} RTT. Let p_n be the probability that the window is reduced in the n^{th} RTT. Then, p_n is given by

$$p_n = 1 - (1-q)^{W_n}. \tag{2}$$

The process A_n is defined as

$$A_n = \begin{cases} \alpha & \text{w.p.} \quad 1-p_n, \\ \beta & \text{w.p.} \quad p_n. \end{cases}$$

Let k be equal to $-\log[\beta]/\log[\alpha]$. Now, we propose an approximation to the above model. For $q \cdot B_u << 1$, we can approximate p_n as

$$p_n \approx qW_n. \tag{3}$$

Using the approximation (3), the average window drop probability, $E[p]$, in an RTT is given by

$$E[p] = qE[W]. \tag{4}$$

In [5], the throughput expression for window independent random losses, when the sender's window is upper bounded, was obtained as

$$E[W] = B_u(1-(k+1)p)\frac{1-\alpha^{-1}}{p\alpha^{-(k+1)} - \alpha^{-1} + (1-p)}, \tag{5}$$

where p was the probability of decreasing the window in an RTT. This probability was independent to W. Eqn. (5) together with (4) gives

$$E[W] = B_u(1-(k+1)qE[W]) \cdot \frac{1-\alpha^{-1}}{qE[W]\alpha^{-(k+1)} - \alpha^{-1} + (1-qE[W])}.$$

The above equation is a quadratic equation in $E[W]$, namely

$$c_2 E[W]^2 + c_1 E[W] + c_0 = 0, \tag{6}$$

where $c_2 = q\frac{1-\alpha^{-(k+1)}}{1-\alpha^{-1}}$, $c_1 = -(1+(k+1)qB_u)$, and $c_0 = B_u$. Therefore, its roots can be explicitly written as

$$E[W]_{1,2} = \frac{-c_1 \pm \sqrt{c_1^2 - 4c_2c_0}}{2c_2}, \tag{7}$$

Proposition 3.1 *The solution of equation (6) which satisfies the inequality $E[W] \leq B_u$ is*

$$E[W] = \frac{-c_1 - \sqrt{c_1^2 - 4c_2c_0}}{2c_2}. \tag{8}$$

We can now obtain an approximate throughput of the session in the window dependent loss model by using (8). We note that this expression is an approximation and we shall compare this approximation with actual simulation results at the end of Section 5.

We can now compute the limit of (8) when B_u tends to infinity. It can be shown that

$$\lim_{B_u \to \infty} E[W] = \frac{1}{q \cdot (k+1)}. \tag{9}$$

The above equation gives the relation between the expected window size (or, equivalently, the expected throughtput) of a connection when the loss probability is sufficiently large such that the upper bound of the window is rarely reached or when there is no upper bound on the window. The expected throughput is given by

$$E[\eta] = \frac{1}{RTT \cdot q \cdot (k+1)} \tag{10}$$

The expected throughput of the MIMD algorithm is inversely proportional to the packet error probability, unlike that of the AIMD algorithm. It is also inversely proportional to k which is the ratio of the logarithm of the increase parameter to the logarithm of the decrease parameter. This result is similar to the result obtained in [4].

4 Markovian Window Independent Losses

4.1 General Model

In this section, we analyze the window process in the presence of Markovian and window independent losses. In each RTT, the probability of experiencing a loss is independent of the window. However, this probability depends on the loss probability in the previous RTT. Let L_n denote the state of the loss process in the n^{th} RTT. The state space of L_n is assumed to be finite with cardinality N. When L_n is in state i, the loss probability in the n^{th} RTT is p_i for $i = 1, 2, ..., N$. Let A denote the transition probability matrix of L_n. The window evolution conditioned on the error process being in state i is given by

$$W_{n+1}|(L_n = i) = \begin{cases} \min(\alpha W_n, B_u) & \text{w.p.} \quad 1 - p_i, \\ \max(\beta W_n, B_l) & \text{w.p.} \quad p_i. \end{cases} \tag{11}$$

In this model we consider the window process to be bounded from both above and below. We now make the transformation

$$Y_n = \frac{\log[W_n] - \log[B_l]}{\log[\alpha]}, \tag{12}$$

to obtain

$$Y_{n+1}|(L_n = i) = \begin{cases} \min(Y_n + 1, M) & \text{w.p.} \quad 1 - p_i, \\ \max(Y_n - k, 0) & \text{w.p.} \quad p_i. \end{cases} \tag{13}$$

The numbers $k = -\frac{\log[\beta]}{\log[\alpha]}$ and $M = \frac{\log[B_u] - \log[B_l]}{\log[\alpha]}$ are assumed to be integers. With this formulation, the state space of Y_n is the set of integers $0, 1, ..., M$, and there is a one to one relationship between W and Y, Therefore, the distribution of W can be obtained from the distribution of Y.

The couple (Y_n, L_n) is a two dimensional discrete-time Markov chain. Let $P = \text{diag}(p_k)$ be a $N \times N$ matrix. Let T be the transition probability matrix of (Y_n, L_n). Then,

$$T = \begin{bmatrix} PA & (I-P)A & 0 & 0 & \dots & 0 \\ PA & 0 & (I-P)A & 0 & \dots & 0 \\ PA & 0 & 0 & 0 & \dots & 0 \\ 0 & PA & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & PA & 0 & 0 & (I-P)A \end{bmatrix}.$$

The matrix T is the transition matrix of a finite $G/M/1$-type Markov chain. The stationary probability vector of the matrix T can be obtained using the numerical techinques [6].

We can obtain more explicit results using matrix-analytic and matrix geometric methods if we consider two particular cases: a) the upper bound B_u is infinity; and b) the lower bound B_l is zero. This is a topic of our future research.

5 Simulation Results

In the congestion avoidance phase, Scalable TCP uses the following algorithm to update the sender's window at the end of every RTT:

$$W_{n+1} = \begin{cases} 1.01 \times W_n & \text{if no losses are detected in the present RTT,} \\ 0.875 \times W_n & \text{if one or more losses are detected in the present RTT.} \end{cases}$$

The simulations are performed using *ns-2* [7]. The source code of standard TCP was modified to incorporate the multiplicative increase of Scalable TCP. The simulation setup has a source and a destination node. The source node has infinite amount of data to send and uses Scalable TCP with New Reno flavor. The link bandwidth is 150Mbps and the two way propagation delay is 120ms. The window at the source is limited to 2000 packets to emulate the receiver advertised window. The BDP for this system is approximately 2250 packets (packet size is 1040 bytes). In the Scalable TCP we have implemented in *ns-2*, the following assumptions are made:

- The minimum window size, B_l, is 8. The growth rate of Scalable TCP is very small for small window sizes. It has been recommended in [4] to use the Scalable algorithm after a certain threshold.
- There is no separate slow start phase since slow start can be viewed as a multiplicative increase algorithm with $\alpha = 2$.
- For each positive ACK received, the window is increased by $\alpha - 1$ packets. When a loss is detected, the window is reduced by a factor of β. α is taken as 1.01 and β is taken as 0.86. This value of β gives $k = -\frac{\log[\beta]]}{\log[\alpha]} \approx 15$. We set α and β in this way so as to be close to the values recommended in [4] ($\alpha = 1.01$, $\beta = 0.875$).

In the simulations, the desired performance metrics are obtained by sampling the window at an interval of $RTT = 0.12s$. We would like to note that in the present setting RTT is very close to the propagation delay, and hence, does not vary much. We compare the approximate throughput formula for the window dependent loss model as obtained using (8), with simulations. In Figures 1 and 2, the throughput is plotted as a function of the packet loss probability, q, for

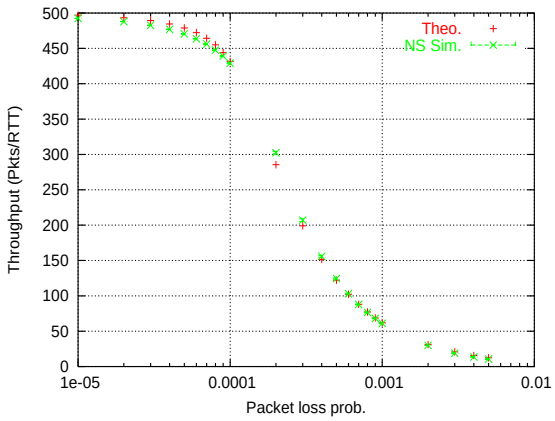

Figure 1: Throughput versus packet loss probability. $B_u = 500$.

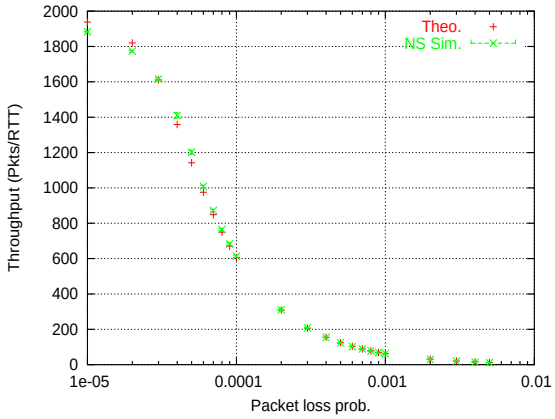

Figure 2: Throughput versus packet loss probability. $B_u = 2000$.

two different maximum window sizes, $B_u = 500$ and $B_u = 2000$. The approximation gives a good match over a large range of values of q. Although we had assumed $qB_u << 1$, the approximation seems to give a good match for larger values of q, too. As q increases, the expected window size, $E[W]$, and the probability of being near B_u decreases. The inequality, $qE[W] << 1$, still holds for larger values of q, and therefore the approximation seems to give a good match.

5.1 Effect of Window Independent Markovian Losses

The effect of bursty losses on the behaviour of AIMD protocols was studied in [8]. It was observed that, in the absence of explicit window limitation, the throughput of TCP improved as the burstiness in the losses increased. In this subsection, we study the behaviour of Scalable TCP when burstiness increases in the losses. First, we describe the error model used, and then compare the result obtained using the model of Section 4 with simulations.

In order to induce burstiness in the losses, we consider a Markovian loss process with two states. In the "GOOD" state, the TCP session observes no losses. In every RTT that the loss process is in the "BAD" state, the session suffers a loss with probability one. We induce burstiness in losses by varying the average duration that the loss process stays in the "BAD" state. In the notation of Section 4,

$$A = \begin{bmatrix} g & 1-g \\ 1-b & b \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{14}$$

The stationary vector of A is given by

$$\gamma = \left[\frac{1-b}{2-b-g} \quad \frac{1-g}{2-b-g} \right]. \tag{15}$$

Let p_a be the average probability of error. Then,

$$p_a = \frac{1-g}{2-b-g}.$$

We denote the burstiness parameter by b. As b increases, the probability that a loss would occur in the following RTT given that a loss occured in the present RTT also increases. Therefore, losses tend to occur more in bursts as b increases. In order to see the effect of burstiness, we fix an average loss probability and vary b from 0 to 0.9. When b is zero, the loss process switches to a good state immediately after a loss. Therefore, losses cannot occur in consecutive RTTs and hence there is no burstiness in losses.

The basic simulation setup is the same as in the independent loss scenario. However, the losses now occur according to a Markovian loss process with transtion matrix A and loss probability matrix P given by (14). In Figure 3, the average loss probability is fixed at 0.02. We observe that the throughput decreases as the burstiness parameter increases. This observation may seem to be contrary to that in [8] where it was noted that bursty losses improve the throughput. However, we note that, unlike in [8], in this study the sender's window is limited by the receiver's advertised window, and this leads to the decrease in throughput.

As the average loss probability increases, the effect of bursty losses also increases. The graph of throughput versus the burstiness parameter for $p_a = 0.04$ is given in Figure 4.

6 Conclusions

First, we presented a mathematical model and an approximate analysis for computing the moments of

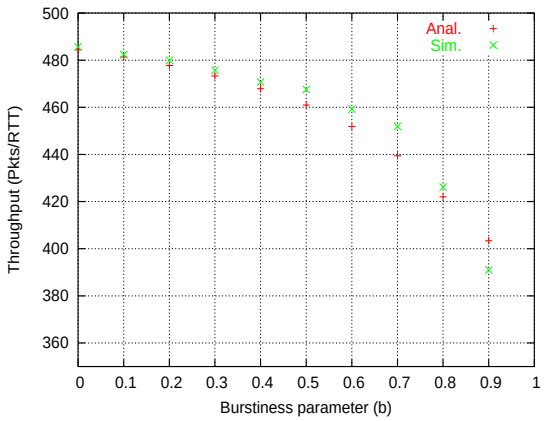

Figure 3: Throughput versus burstiness parameter. $p_a = 0.02$. $B_u = 500$.

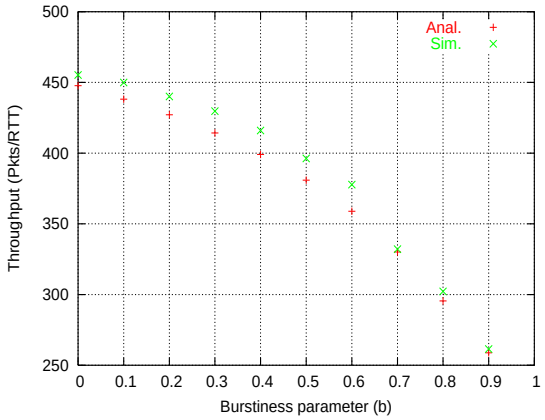

Figure 4: Throughput versus packet loss probability. $p_a = 0.04$. $B_u = 500$.

the window size, and, in particular, the throughput of a single connection using MIMD congestion control algorithm in the presence of random window dependent losses. In the case of a limitation on the sender's window, the throughput could be computed by solving a quadratic equation. In the absence of a limitation, the throughput was proportional to inverse of the loss probability. We then studied the window behaviour in the presence of Markovian window independent losses. It was observed that when the sender's window is limited by the receiver's advertised window, bursty losses lead to decrease in the throughput.

References

[1] S. Floyd. HighSpeed TCP for Large Congestion Windows. RFC 3649, Experimental, December 2003.

[2] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: Motivation, Architecture, Algorithms, Performance. In *Proceedings of the IEEE INFOCOM*, March 2004.

[3] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control (BIC) for Fast Long-Distance Network. In *Proceedings of the IEEE INFOCOM*, March 2004.

[4] T. Kelly. Scalable TCP: Improving Performance in Highspeed Wide Area Networks. *Computer Comm. Review*, 33(2):83–91, April 2003.

[5] E. Altman, K. Avratchenkov, C. Barakat, A. A. Kherani, and B. J. Prabhu. Analysis of Scalable TCP. In *Proceedings of the IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC)*, 2004.

[6] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability. SIAM, 1999.

[7] S. McCanne and S. Floyd. ns: Network Simulator. Available at http://www.isi.edu/nsnam/ns/.

[8] E. Altman, K. E. Avrachenkov, and C. Barakat. TCP in Presence of Bursty Losses. In *Proceedings of ACM SIGMETRICS*, 2000.