On Throughput Stabilization of Network Transport

Nageswara S. V. Rao, Qishi Wu, and S. Sitharama Iyengar

***Abstract*—A number of network applications require stable transport throughput for tasks such as control and coordination operations over wide-area networks. We present a window-based method that achieves stable throughput at a target level by utilizing a variation of the classical Robbins-Monro stochastic approximation algorithm. We analytically show the stability of this method under very mild conditions on the network, which are justified by Internet measurements. Our User Datagram Protocol (UDP)-based implementation provides stable throughput over the Internet under various traffic conditions.**

***Index Terms*—Robbins-Monro algorithm, stochastic approximation, transport stabilization.**

I. Introduction

A number of next generation network-based applications require stable throughput for control of processes over wide-area networks. Examples include controlling mobile robots remotely over wide-area networks, computational steering of remote large scientific simulations, and coordinated visualization of distributed datasets. Typically, the bandwidth requirements are modest in these applications, often requiring only a small fraction of the available bandwidth. But, it is extremely important that the throughput rate at remote site(s) be stable in presence of dynamic traffic conditions. For example, large amount of jitter in throughput can destabilize the control loops needed for remote robots, possibly causing severe damage to them.

The widely deployed Transmission Control Protocol (TCP) is not designed for providing such stable throughput for several reasons. First, it continues to increase its throughput until losses are encountered which often results in much higher throughput than needed, especially in over-provisioned networks. Second, perhaps more importantly, TCP drastically reduces its throughput to levels significantly below the required in response to bursty losses. While TCP throughput can be curtailed around the desired values by clipping the flow windows, its nonlinear dynamics make this task very challenging [6]; in fact, TCP is provably chaotic under certain conditions [5], which makes the throughput stabilization particularly difficult. Even if such approach is successful, TCP offers no protection against the "underflow" particularly in high traffic networks.

We show that a dynamic version of the classical Robbins-Monro method [3], [7] offers a provably stable throughput under very general conditions that can be justified using Internet measurements. Our implementation based on User Datagram Protocol (UDP) achieved very stable and robust throughput over the Internet under various traffic conditions.

II. Problem Formulation

We consider the problem of stabilizing a transport stream from a source node S to a destination node D over a wide-area network, typically the Internet. The objective is to achieve a *target* throughput rate γ at D by dynamically adjusting the sending rate at S in response to network conditions. Packets are sent from S and are acknowledged by D. Both packets and their acknowledgments can be delayed or lost during the transmission due to a variety of reasons, including buffer occupancy levels at routers and hosts, and link level losses. Let $R_S(t)$ and $G_D(t)$ denote the *sending rate* at S and *throughput* or *goodput* at D, respectively. Each rate is given by the number of packets sent and received at S and D, respectively, divided by the interval duration. The *response plot* corresponds to values of $G_D(.)$ plotted against $R_S(.)$. Typically, in wide-area transports, $G_D(.)$ increases with $R_S(.)$ in an overall sense for lower values, often incurring very low losses. For higher values of $R_S(.)$, there is an increase in losses and $G_D(.)$ decreases in an overall sense. Such overall behavior is well-known and has been ideally modeled as fixed smooth concave utility functions in a number of transport control works [2], [4]; under these assumptions, the stabilization problem involves simply computing R_S value such that $G_D(t) = \gamma$ (if such utility functions are known precisely).

In practical wide-area networks, however, such precise characterization is not feasible. In most cases, one only has access to the measurements at source (including the ones sent by destination) to adjust $R_S(t)$. The measurements shown in Fig. 1 are collected between Oak Ridge National Laboratory (ORNL), Oak Ridge, TN, and Louisiana State University (LSU), Baton Rouge, LA. The datagrams within a window are all sent continuously followed by a waiting period, and this process is repeated. In the horizontal plane each point corresponds to window-size and waiting-time (or idle-time) pair, the ratio of which specifies $R_S(t)$; the top and bottom plots represent $G_D(t)$ and *loss rate*, respectively. For illustration, let us fix the waiting-time and increase the window-size which corresponds to taking vertical slices of the plots parallel to the window-size axis. There are two important features: 1) there is an overall trend of increase followed by decrease in G_D as R_S is increased; this overall behavior is quite stable although the transition points vary over time and 2) the plot is quite nonsmooth mostly because of the randomness involved in packet delays and losses; derivation of smooth utility functions from the response plots is inherently approximate and requires a large number of observations in a

Manuscript received May 8, 2003. The associate editor coordinating the review of this letter and approving it for publication was Dr. J. Choe. This work was supported by the Defense Advanced Research Projects Agency under Grant MIPR K153, by the National Science Foundation under Award ANI-0335185 and Award ANI-0229969, and by Engineering Research Program and High-Performance Networking Program of Office of Science, U. S. Department of Energy under Contract DE-AC05-00OR22725 with UT-Battelle, LLC.

N. S. V. Rao and Q. Wu are with Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6364 USA (e-mail: raons@ornl.gov; wuqn@ornl.gov).

S. S. Iyengar is with Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: iyengar@bit.csc.lsu.edu).

Digital Object Identifier 10.1109/LCOMM.2003.822508

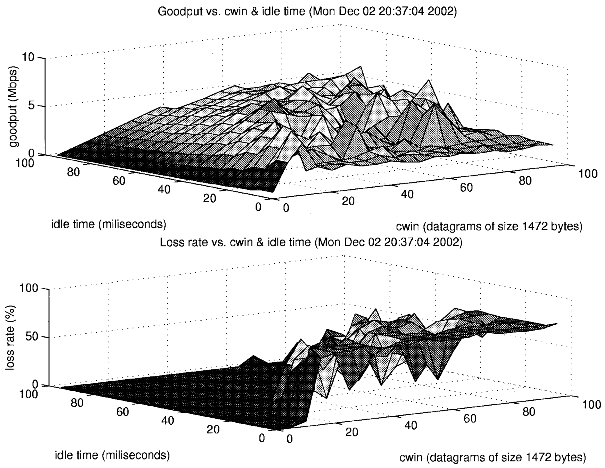

Fig. 1. Internet measurements with sending rate along horizontal plane, and throughput and loss rate along vertical axes in top and bottom plots, respectively.

preprocessing step. In practice, S computes an approximation $\hat{G}_S(t)$ to $G_D(t)$ typically based on acknowledgments.

We assume that γ is specified much below the peak of the response plot, where the loss rate is very small (if any) as shown in Fig. 1. Due to a wide variety of factors such as competing traffic, buffer occupancy at hosts and routers, and packet losses, $G_D(t)$ has a significant random component in addition to its overall dependence on $R_S(.)$. Furthermore, S does not have access to $G_D(t)$ and it's estimates are communicated over the network, either directly or indirectly as part of acknowledgments. Consequently, $\hat{G}_S(t)$ is a further "randomized" version of $G_D(t-\delta)$, for some $\delta > 0$.

Let $\Gamma(r)$ be the *response regression* given by the expected value of G_D corresponding to fixed $R_S(t) = r$, that is

$$E\left[G_D(t)|R_S(t) = r\right] = \Gamma(r).$$

Let the *stabilization rate* r_γ be given by $\Gamma(r_\gamma) = \gamma$. Here γ is chosen such that r_γ is within the initial increasing part of $\Gamma(.)$. We assume that $\Gamma(r)$ is *locally monotonic* in the neighborhood of γ such that: $\Gamma(r) > \gamma$ for $r > r_\gamma$ and $\Gamma(r) < \gamma$ for $r < r_\gamma$. This assumption is consistent with the measurements in Fig. 1 and, also with the concavity assumptions in [2], [4]. Note that $G_D(.)$ is typically nonsmooth and $\Gamma(.)$ is not known.

III. Throughput Stabilization

In our method $R_S(t)$ is adjusted at each time t_i an acknowledgment is received, by using an estimate $\hat{G}_S(t)$ of $G_D(t)$. At time t_i, let the *window-size* W_i denote the number of packets to be sent followed by a *waiting-time* τ_w such that for $t_i \le t < t_{i+1}$,

$$R_S(t) = \frac{W_i}{\tau_a + \tau_w}$$

where τ_a is the time needed for transmitting the packets at S. At $t = t_{i+1}$, the window size is updated as follows:

$$W_{i+1} = W_i - \rho_i\left[\hat{G}_S(t_i) - \gamma\right] \tag{3.1}$$

where $\rho_i = (k(\tau_a + \tau_w)/i^\alpha)$ for $0.5 < \alpha < 1$ and $k > 0$ a suitably chosen constant. This method is a specific form of the well-known stochastic approximation (SA) algorithm [3]. Intuitively, W_i is increased if the estimate of throughput is below γ and decreased otherwise. Initially W_0 is chosen based on measurements so that $R_S(t_0)$ is in the vicinity of γ.

We compute the *acknowledgment rate* $\hat{A}_S(t)$ with every acknowledgment received at S by dividing the total number of received acknowledgments by the time expired so far. Thus $R_S(t) - \hat{A}_S(t)$ is the sum of loss rate of packets from S to D and acknowledgments from D to S. We assume a *symmetric loss* process so that the loss rate of packets and that of acknowledgments are identically given by $[R_S(t) - \hat{A}_S(t)]/2$. Then we employ the estimate

$$\hat{G}_S(t) = \hat{A}_S(t) + \frac{\left[R_S(t) - \hat{A}_S(t)\right]}{2} = \frac{\left[\hat{A}_S(t) + R_S(t)\right]}{2}.$$

If $R_S(.)$ is fixed, the symmetric loss condition yields the flow equation $E[R_S(t)] = E[G_D(t) + (R_S(t) - \hat{A}_S(t))/2]$ or equivalently $E[R_S(t)] = 2E[G_D(t)] - E[\hat{A}_S(t)]$. Then, using the above equation, it follows that $\hat{G}_S$ is a conditionally consistent estimator of G_D, namely

$$E\left[\hat{G}_S(t)|G_D(t) = x\right] = x.$$

For the nonsymmetric loss case, the acknowledgments can be augmented with the one-way loss estimator from D, which can be used to obtain a consistent estimator of G_D.

We now show the stability of (3.1) using Theorem 2 from [1], namely $E[(R_S(t_i) - r_\gamma)^2] \to 0$ as $i \to \infty$. To show this (3.1) can be rewritten as

$$W_{i+1} = W_i - a_i\left[\hat{Y}_S(t_i) - \gamma(\tau_a + \tau_w)\right] \tag{3.2}$$

where we denote $\hat{Y}_S(t) = \hat{G}_S(t)(\tau_a + \tau_w)$ and $a_i = k/i^\alpha$. Then we have

$$E\left[\hat{Y}_S(t_i)|W_1, W_2, \ldots, W_i\right] = (\tau_a + \tau_w)\Gamma\left(\frac{W_i}{\tau_a + \tau_w}\right).$$

We assume that

$$\mathrm{var}\left[\hat{Y}_S(t_i)|W_1, W_2, \ldots, W_i\right] \le \sigma^2 \tag{3.3}$$

for some σ, which can be easily ensured by thresholding the W_i values. We assume there exist k_0 and k_1 such that

$$k_0|r - r_\gamma| \le |\Gamma(r) - \gamma| \le k_1|r - r_\gamma| \tag{3.4}$$

which is justified by the measurements at low $R_S(.)$ values. Let $\omega_\gamma = r_\gamma(\tau_a + \tau_w)$ correspond to the ideal window size that achieves the stabilization rate γ in an average sense. For a constant value of ω_γ, we invoke Theorem 2 of [1] for the case it is slowly varying; note that ω_γ is an average quantity, and the measured throughput levels for this window could be quite varied. Then from [1] we have the following stability result:

Theorem 1: Under the locally monotonic response regression, symmetric loss and conditions in (3.3) and (3.4), the window-sizes computed in (3.2) satisfy the stability conditions $E[(W_i - \omega_\gamma)^2] = O(i^{-\alpha})$ or equivalently $E[(R_S(t_i) - r_\gamma)^2] = O(i^{-\alpha})$.

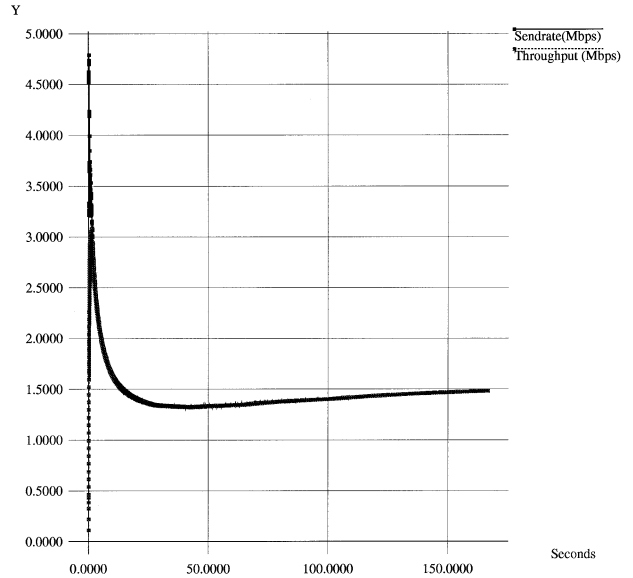

Fig. 2. Stabilization at 1.5 Mb/s.

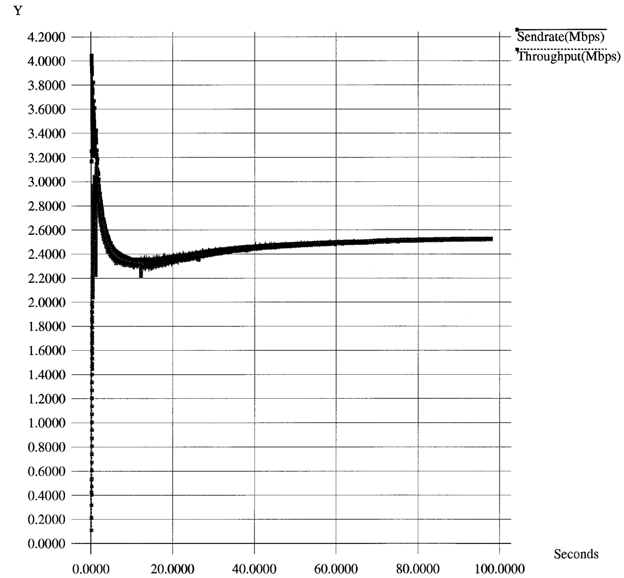

Fig. 3. Stabilization at 2.5 Mb/s throughput under background large file ftp at host with web browsing.

Since $\hat{G}_S(t)$ is a random quantity, it is very critical that the step size ρ_i in (3.1) be chosen to satisfy the classical Robbins-Monro conditions [7]: (i) $\rho_i \rightarrow 0$ as $i \rightarrow \infty$, (ii) $\sum_{i=1}^{\infty} \rho_i = \infty$, and (iii) $\sum_{i=1}^{\infty} \rho_i^2 < \infty$. In particular, a fixed-step size (e,g, used in [4] for smooth deterministic case) does not guarantee stability, and in fact did not stabilize in our experiments over the Internet.

The above stability analysis remains valid for the case window-size is fixed and waiting-time is adapted in a manner similar to (3.1). Our experimental results are qualitatively identical in both these cases.

For a fixed γ, changes in $\Gamma(.)$ will result in different r_γ (or ω_γ) values. The algorithm in (3.1) is very robust in presence of changes in r_γ. Changes in network traffic (for example due to new connections and/or termination of old ones), typically have very small effect on the regression function $\Gamma(.)$. In particular, the effect of individual traffic streams is mitigated, resulting in very small changes in r_γ in the low loss region. Indeed, the results of [1] are valid when r_γ is time varying. If $r_\gamma(t_{i+1}) - r_\gamma(t_i) = O(i^{-\omega})$ the algorithm in (3.1) stabilizes as:

$$E\left[(R_S(t_i) - r_\gamma)^2\right] = \begin{cases} O(i^{-\alpha}) & \text{if } \omega \geq \frac{3}{2}\alpha \\ O\left(i^{-[2(\omega-\alpha)]}\right) & \text{if } \omega < \frac{3}{2}\alpha. \end{cases}$$

IV. Experimental Results

Our method is implemented using UDP, and tested extensively between ORNL and LSU. ORNL is connected to ESnet which peers with Abilene network in New York (before recent upgrade). Abilene runs from New York via Washington, DC, and Atlanta, GA, to Houston, TX, where it connects to LSU via a regional network. In terms of network distance, these two sites are separated by more than two thousand miles, and both ESnet and Abilene have significant traffic. Fig. 2 shows typical results for target throughput at 1.5 and 3.0 Mb/s all below the peak bandwidth but above throughput 1.09 Mb/s achieved by default TCP. In each plot, top and bottom curves correspond to $R_S(t)$ and $G_D(t)$, respectively, which often overlap indicating low loss conditions. The stabilization typically occurred under very low albeit nonzero packet loss. The throughput was remarkably robust and was virtually unchanged when ftp transfers of various file sizes were made at the local LAN host together with various web browsing operations as shown in Fig. 3.

V. Conclusions

This letter establishes that the classical statistical control methods provide provably robust throughput stabilization over wide-area networks, and our implementations support these results over the Internet. However, in practice the target throughput is meant only to be a small fraction of available bandwidth so as to minimally affect the other traffic streams. In a certain sense, this approach is a new way of designing stable transport protocols wherein the inherent randomness in the measurements is handled by the SA algorithm. Topics of future interest include throughput maximization using SA algorithms (such as Kiefer–Wolfowitz or simultaneous perturbation [3]) and their ability to be fair to other traffic streams in the network.

References

[1] V. Dupac, "A dynamic stochastic approximation method," *Annals of Mathematical Statistics*, vol. 36, no. 6, pp. 1695–1702, 1965.

[2] F. P. Kelly. Mathematical modeling of the internet. presented at *Proc. Int. Congress on Industrial and Applied Mathematics*. [Online]Available: http://www.statslab.cam.ac.uk/~frank/mmi.html

[3] H. J. Kushner and C. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[4] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Syst. Mag.*, 2002.

[5] N. S. V. Rao and L. O. Chua, "On dynamics of network transport protocols," in *Proc. Workshop on Signal Processing, Communications, Chaos and Systems*, 2002, pp. 29–43.

[6] N. S. V. Rao and W. Feng, "Performance trade-offs of tcp adaptation methods," in *Int. Conf. on Networking*, 2002, pp. 499–510.

[7] M. T. Wasan, *Stochastic Approximation*. Cambridge, UK: Cambridge Univ. Press, 1969.