THE BEHAVIOR OF CERTAIN STOCHASTIC PROCESSES ARISING IN WINDOW PROTOCOLS

S. Savari and E. Telatar
600 Mountain Avenue, Lucent Technologies Bell Laboratories
Murray Hill, NJ 07974, USA

Abstract

Window based network flow control protocols, such as TCP, modulate the number of unacknowledged packets the protocol is allowed to have outstanding. Such protocols change the window size when they receive positive or negative acknowledgments, where the latter kind may be inferred from timeouts. Together with a communications channel that loses packets at random, such a protocol induces a stochastic process on the window size. In this paper we consider a broad class of window based protocols, and analyze various statistics of the induced stochastic process. We demonstrate that all these protocols can be treated analytically using the theory of semi-Markov processes.

1 Introduction

The purpose of network flow control protocols is to try to adapt the rate of packet transmission to the prevailing network characteristics. Window based protocols attempt to achieve this goal by allowing the transmission of a certain number of new packets before all existing packets are acknowledged and then updating this allowed number when acknowledgments are received. The set of packets that can be transmitted without acknowledgments is called the *window*, and each window based protocol specifies a set of rules to change the size of this window. We will refer to the implementation of a protocol as an algorithm, and sometimes ignore the distinction between the two.

Consider a typical window based flow control algorithm: The algorithm is in either a *slow-start* or a *congestion avoidance* mode. If it is in the congestion avoidance mode, the algorithm grows its window size when it detects that a packet has been successfully transmitted. The amount of the growth may depend on the current window size. The growth in window size can be fractional, in which case the window grows physically by one for every so many packets. If a packet loss is detected, the algorithm enters the slow-start mode, sets the window size to some small number and for each detection of a successful transmission grows the window size; the growth rate may again depend on the window size. Slow-start mode ends when the window size reaches a certain threshold, at which point the algorithm enters the congestion avoidance mode. The threshold may depend, for example, on the size of the window when the packet loss was detected.

The state of such a protocol can be described by specifying the mode and the window size. If the algorithm is in the slow-start mode, the threshold window size is needed to complete the state description; in the congestion avoidance mode we can set the threshold to infinity. It is possible to summarize this information more compactly with only two variables, the window size and an auxiliary variable. Our convention will be that if the window size is less than the auxiliary variable then we are in slow-start mode and the auxiliary variable has the meaning of the threshold window size; otherwise, we are in congestion avoidance mode and the auxiliary variable does not have an operational meaning.

To recapitulate, we can represent the state $\vec{X}$ of the process at a given time by an ordered pair of real numbers:

$$\vec{X}(t) = \big(W(t), Z(t)\big).$$

The first member of the pair, $W(t)$, will be the current window size. The second member, $Z(t)$, is an auxiliary variable that indicates whether the window growth is in slow-start mode or congestion avoidance mode: the cases $W(t) < Z(t)$ and $W(t) \geq Z(t)$ correspond to the former and the latter modes, respectively.

To complete the description of the communication system we need to model the packet losses. We will assume that the packet losses occur according to a Poisson process whose rate at a given time t depends on the state of the protocol at that time $\vec{X}(t)$.

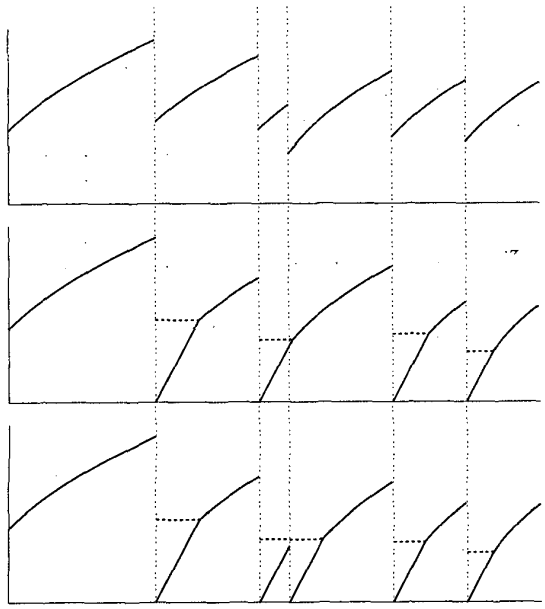
The evolution of $\vec{X}(\cdot)$ in time is governed by a stochastic differential equation:

$$\vec{X}(t+dt) = \begin{cases} \alpha\big(\vec{X}(t)\big) & \text{w. p. } \lambda(\vec{X}(t))dt \\ \vec{X}(t) + \beta\big(\vec{X}(t)\big)dt & \text{w. p. } 1 - \lambda(\vec{X}(t))dt \end{cases}$$

where $\lambda(\vec{X}(t))$ is the packet loss detection rate when the state of the process is $\vec{X}(t)$. Ordinarily, one will have an $\alpha(\cdot)$ such that the window size is reduced when a packet loss is detected, and a $\beta(\cdot)$ which increases the window size when an acknowledgment is received.

The formulation above is very general. In this paper, however, we will restrict the model in a number of ways. We will assume that $\lambda(\cdot)$ depends on $\vec{X}$ only through is first component W. Also, we will focus on models in which the auxiliary variable Z stays constant during a period where no packet loss is detected; i.e., we will only consider functions β whose second component is identically zero. Note that other choices of β allow an evolution of Z between successive packet loss detections. This would allow for a more complex interplay between the probability of a packet loss and the history of the process. Even with the above restrictions this formulation allows us to accommodate a larger

0-7803-5796-5/99/$10.00 © 1999 IEEE



The figures show the evolution in time of the window size and the threshold for various protocols. From top to bottom the figures correspond to Examples 1–3. The packet loss instants are identical for all charts and are indicated by the vertical dashed lines. The threshold is indicated by the horizontal dotted lines when a threshold is active.

Figure 1: Comparison of windowing protocols

class of protocols and a larger class of packet-loss models than formulations considered before.

To motivate this representation let us give some examples. The reader may find it useful to refer to Figure 1.

EXAMPLE 1. The model can accommodate a process that has no slow-start mode: consider a process that halves the window size each time a loss is detected and evolves according to the differential equation $W' = 1/W$ in between successive packet loss detections. Since α governs the system in the event of a packet loss detection, and β governs the system during periods of no packet loss detection, taking

$$\alpha\big((w,z)\big) = (w/2, 0),\ \beta\big(w,z\big) = (1/w, 0),\ \text{and}\ Z(0) = 0$$

will model this process. Observe that this formulation is consistent with no slow-start phase because $Z(t)$ is always zero. This special case is closely related to the model studied in [1]. Note, however, that by allowing the loss rate to depend on the window size our model generalizes that of [1]. It can be argued that this particular formulation is a good model of TCP-Reno since Reno's fast recovery shortens the duration of slow-start periods.

EXAMPLE 2. A more realistic analysis will include the effects of the slow-start mode. The simplest such model will include a slow-start mode that ignores packet losses detected during slow start. That is, whenever a packet loss is detected in slow-start mode, the state $\vec{X}$ does not change. If a packet loss is detected in congestion avoidance mode, the window size is set to zero and the auxiliary variable is set to half the window size just prior to the packet loss detection; the auxiliary variable remains fixed until the next congestion avoidance phase begins. The function

$$\alpha\big((w,z)\big) = \begin{cases} (w,z) & w < z \\ (0, w/2) & w \geq z \end{cases}$$

provides this behavior. If the window size is to increase linearly during slow-start and obey $W' = 1/W$ during congestion avoidance, then $\beta(\cdot)$ satisfies

$$\beta\big((w,z)\big) = \begin{cases} (1,0) & w < z \\ (1/w, 0) & w \geq z. \end{cases}$$

EXAMPLE 3. We next consider another algorithm that is closer to actual implementations of window based control. In this protocol, the window size is set to zero every time a packet loss is detected. The auxiliary variable is modified only for losses detected during congestion avoidance:

$$\alpha\big((w,z)\big) = \begin{cases} (0,z) & w < z \\ (0, w/2) & w \geq z \end{cases}$$

$$\beta\big((w,z)\big) = \begin{cases} (1,0) & w < z \\ (1/w, 0) & w \geq z. \end{cases}$$

As we will see later in the paper, each of the above algorithms lends itself to an analysis based on the theory of semi-Markov processes. Even though the state process $\vec{X}(t)$ is Markovian, the window size $W(t)$ is, in general, not. Nonetheless, $W(t)$ is a semi-Markov process. Namely, we can identify epochs $S_0, S_1, \ldots$, such that $W_n = W(S_n^+)$ is a discrete time Markov process, and the transition times $S_{n+1} - S_n$ are random, but depend only on W_n and W_{n+1}. For each of the above algorithms one can take as the epochs the instances the algorithm begins a congestion avoidance phase. As we shall see, the structure of a transition period is different for different algorithms. Nonetheless, each protocol can be generalized without changing the structure of the transition periods. In Example 1, the case without slow start, we can modify α so that the window size is reset to an arbitrary increasing function of the window size just before the packet loss detection. Similarly, β can be modified so that the evolution of the process between successive packet loss detections is governed by a prespecified time-invariant differential equation:

$$\alpha\big((w,z)\big) = (\alpha_0(w), 0)$$

$$\beta\big((w,z)\big) = \big(\beta_0(w), 0\big).$$

[2] considers a special case of this protocol with $\alpha_0(w) = w/\gamma$, $\gamma > 1$, $\beta_0(w) = w^m$ with $m < 1$, and the loss rate to independent of the window size.

In Example 2, the case with a slow-start mode in which loss detections are ignored, one can generalize the choice of auxiliary variable and the two differential equations that govern window evolution between successive loss detections:

$$\alpha\big((w,z)\big) = \begin{cases} (w,z) & w < z \\ (0, \alpha_{ca}(w)) & w \geq z \end{cases}$$

$$\beta\big((w,z)\big) = \begin{cases} (\beta_{ss}(w), 0) & w < z \\ (\beta_{ca}(w), 0) & w \geq z \end{cases}$$

When a packet loss is detected during the slow-start phase, the algorithm makes no change in the values of $W(t)$ and $Z(t)$. On the other hand, if a packet loss is detected during the congestion avoidance phase, the algorithm will set $Z(t)$ to a function α_{ca} of $W(t)$ and reset $W(t)$ to zero. During a slow-start period, $W(t)$ will obey $W'(t) = \beta_{ss}(W(t))$ and during a congestion avoidance period, $W(t)$ will obey $W'(t) = \beta_{ca}(W(t))$.

Similarly, Example 3 can be generalized so that

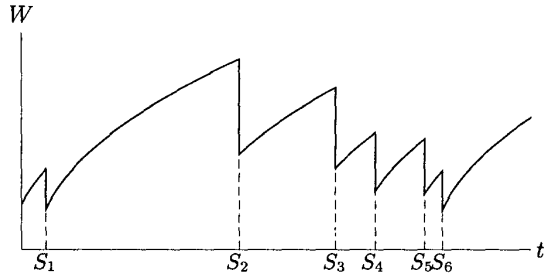
$$\alpha\big((w,z)\big) = \begin{cases} (0,z) & w < z \\ (0, \alpha_{ca}(w)) & w \geq z \end{cases}$$

$$\beta\big((w,z)\big) = \begin{cases} (\beta_{ss}(w), 0) & w < z \\ (\beta_{ca}(w), 0) & w \geq z. \end{cases}$$

A note on how the passing of time is measured is in order before we embark on further details. In the protocols we consider, the state changes only when the protocol receives a positive or negative acknowledgment. This gives rise to a virtual time, which advances by one unit every time such an acknowledgment is received. Accordingly, when we analyze the behavior of the associated stochastic processes in *time*, we are not referring to *real* time but to this virtual time. A further comment is necessary on this point: since all our stochastic differential equations are written in this virtual time, the rate of the variable rate Poisson process that drives the packet losses is with respect to this virtual time and care needs to be taken to convert the packet loss rate with respect to real time to a rate measured in virtual time. For the details of the conversion between real time and virtual time the reader is referred to [3].

2 Analysis

We will consider each of the three types of protocols. As we will see, the analysis of the second and third algorithms builds upon the first one, i.e., the case where is no slow-start mode. The outline of the analysis for the no slow-start mode protocol is as follows: If we take the sequence of values the window size takes just after each packet loss, we observe that they form a Markov chain. Furthermore since between packet losses the evolution of the process is deterministic, it is sufficient to analyze the properties of this Markov chain to capture the behavior of the process at all times. Accordingly we proceed by computing the transition probability density of the chain and finding its steady state distribution. We then compute the statistics of the continuous time process from the statistics of the discrete time chain.



The figure illustrates a sample path of the process with $\alpha_0(w) = w/2$, $\beta_0(w) = 1/w$, $\lambda(w) \equiv 1$. This corresponds to idealized TCP with packet loss probability independent of the window size.

Figure 2: A sample path of the process described by (1)

2.1 No Slow-Start

In this case $Z(t)$ is identically zero, and the state information consists only of the window size $W(t)$. The window size evolves according to

$$W(t+dt) = \begin{cases} \alpha_0(W(t)) & \text{w.p. } \lambda(W(t))\,dt, \\ W(t) + \beta_0(W(t))\,dt & \text{w.p. } 1 - \lambda(W(t))\,dt, \end{cases} \tag{1}$$

Call the instances of time for which the first alternative in (1) applies *crashes*. The process W thus evolves in the following way: between successive crashes, the process increases, evolving according to the first order time-independent differential equation

$$\frac{dW}{dt} = \beta_0\big(W(t)\big), \tag{2}$$

until it is operated on by α_0 at a crash (see Figure 2).

For the models considered in [2] the probabilistic behavior of $W(t)$ at a random point in time is found. The analysis in [2] heavily relies on the assumption that the crash probability is independent of the window size. Our approach to investigating $W(t)$ is new. We derive the steady-state behavior of the window size at crash points. From this we can not only obtain the probability distribution function of the window size at a random point, but also at non-

random points such as just before a crash, just after a crash, etc.

Set $S_0 = 0$, and let S_i denote the i^{th} crash epoch. Let $W_n = W(S_n^+)$, $n \geq 0$ denote the value of the window size just after the n^{th} crash. Observe that, conditional on $W(t_0) = w_0$, the event that there are no crashes in $(t_0, t_0 + t)$ is equivalent to the event that a non-homogeneous Poisson process with rate $\lambda(W(t_0 + \tau))$ in the interval $0 \leq \tau \leq t$ has no arrivals in $(0, t)$. From this observation it follows that

$$P(W_{n+1} > w_{n+1} | W_n = w_n) = \exp\Big(-\int_{w_n}^{\alpha_0^{-1}(w_{n+1})} \frac{\lambda(w)}{\beta_0(w)}\, dw\Big), \quad w_{n+1} \geq \alpha_0(w_n). \tag{3}$$

Let π denote the steady-state density of W_n. Before computing π, let us first evaluate the steady-state time average of a function Φ of W:

$$\overline{\Phi} = \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^\tau \Phi(W(t))\, dt.$$

With a renewal theory argument it can be shown that

$$\overline{\Phi} = \frac{E[\Phi(\alpha_0^{-1}(W_n))/\lambda(\alpha_0^{-1}(W_n))]}{E[1/\lambda(\alpha_0^{-1}(W_n))]}, \tag{4}$$

where the expectations are with respect to the steady-state measure. Observe that this relation allows us to express the time average of any function of the process $W(\cdot)$ in terms of the steady state distribution of the process just after a crash.

2.1.1 Computing the Stationary Distribution

It now remains to compute π so that the expectations in (4) can be evaluated. We have been able to determine π only in the case when $\alpha_0(w) = w/\gamma$ for some $\gamma > 1$, and $\lambda(w)/\beta_0(w) = \mu_1 w^m$ for some real $m > -1$. In this case one can show that

$$\pi(w) = \mu_1 w^m \rho\Big(\frac{\mu_1 w^{1+m}}{1+m}\Big). \tag{5}$$

where $\rho(x) = \sum_{k=0}^\infty a_k \exp(-\kappa^{k+1} x)$, $\kappa = \gamma^{1+m}$, and the generating function $A(z) = \sum_{k \geq 0} a_k z^k$ is given by $A(z) = a_0 \prod_{k \geq 0}(1 - z/\kappa^k)$ with $a_0 = \kappa / \prod_{k>0}(1 - \kappa^{-k})$.

We can also obtain the steady-state time average density $\xi(\cdot)$ of the process $W(\cdot)$ as

$$\xi(w) = c^{-1} \frac{\pi(w/\gamma)}{\lambda(w)} \tag{6}$$

where $c = \gamma \int_0^\infty [\lambda(\gamma w)]^{-1} \pi(w)\, dw$.

The moments of W_n are given by the following: When r is not a negative integer multiple of $m + 1$,

$$E[W_n^r] = \Big(\frac{1+m}{\mu_1}\Big)^{r/(1+m)} \Gamma\Big(\frac{1+m+r}{1+m}\Big) \gamma^{-r} \cdot \prod_{k>0} \frac{1 - \gamma^{-(1+m)k-r}}{1 - \gamma^{-(1+m)k}}. \tag{7}$$

When r is a negative integer multiple of $m + 1$,

$$E[W_n^r] = \Big(\frac{1+m}{\mu_1}\Big)^{r/(1+m)} \frac{(1+m)\gamma^{-r} \log \gamma}{(1 - r/(1+m))!} \cdot \prod_{k=1}^{-1-r/(1+m)} (\gamma^{(m+1)k} - 1). \tag{8}$$

2.1.2 Generalized TCP/IP

As an application of the development above, recall the initial version of Example 1, where the window size is halved upon detecting a packet loss and the window growth rate is inversely proportional to the window size during periods of no packet loss detection. Consider a packet loss detection rate which is proportional to the window size, $\lambda(W(t)) = \lambda_1 W(t)$. Then

$$W(t + dt) = \begin{cases} W(t)/2 & \text{w.p. } \lambda_1 W(t)\, dt, \\ W(t) + \dfrac{\beta_1}{W(t)}\, dt & \text{w.p. } 1 - \lambda_1 W(t)\, dt, \end{cases}$$

The steady-state density of W_n and $W(\cdot)$ are then given by

$$\pi(w) = \frac{\lambda_1}{\beta_1} w^2 \sum_{k=0}^\infty a_k \exp(-8^{k+1} \frac{\lambda_1}{3\beta_1} w^3),$$

$$\xi(w) = Cw \sum_{k=0}^\infty a_k \exp(-8^k \frac{\lambda_1}{3\beta_1} w^3).$$

with $C = \frac{3}{8}\Big(\frac{\lambda_1}{3\beta_1}\Big)^{2/3} \prod_{k>0} \frac{1 - 2^{-3k+1}}{1 - 2^{-3k}}$, and

$$a_0 = \frac{8}{\prod_{k>0} 1 - 8^{-k}}, \quad a_k = a_{k-1} \frac{8}{8^k - 1}, \quad k > 0.$$

It also follows that

$$\overline{W(t)} = \Big(3\frac{\beta_1}{\lambda_1}\Big)^{1/3} \Gamma(2/3)^{-1} \prod_{k \geq 0} \frac{1 - 2^{-3k-3}}{1 - 2^{-3k-2}} \doteq 1.26547 \Big(\frac{\beta_1}{\lambda_1}\Big)^{1/3}.$$

2.2 Loss Ignoring Slow-start

In this section we will consider algorithms that admit a slow-start phase but ignore packet losses during such a phase. Consider the instances S_0, S_1, ... when the protocol begins a congestion avoidance phase. The window size $W_n = W(S_n)$ at these instances form a Markov process. Since the protocol changes neither the window size

nor the threshold for packet losses that are detected during a slow-start phase, the transition probabilities for the Markov process W_n are identical to the ones considered in the last section.

Observe that the interval between successive starts of a congestion avoidance phase consists of a congestion avoidance interval followed by a slow-start interval which is a deterministic function of the window size when the next congestion avoidance phase begins. This allows us to combine renewal theory with the results of the previous section to obtain the statistical properties of this protocol.

We specialize to the case where $\lambda(w)/\beta_{ca}(w) = \mu_1 w^m$ for some real $m > -1$. Also assume that $\beta_{ss}(w) = \beta_2 w^{-c}$ for $c \geq 0$. For $\Phi(w) = w^f$, and $\lambda(w) = \lambda_1 w^\ell$, when $c > -1 - f$,

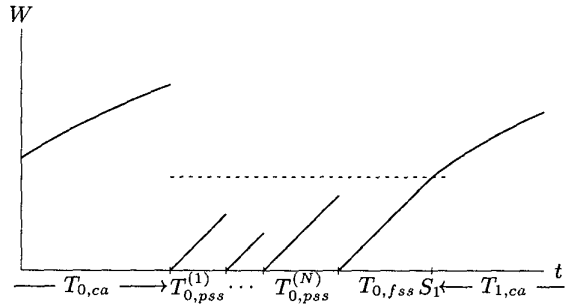
$$\overline{\Phi} = \frac{\frac{\gamma^{f-\ell}}{\lambda_1} E[W_n^{f-\ell}] + \frac{1}{\beta_2(1+c+f)} E[W_n^{1+c+f}]}{\frac{\gamma^{-\ell}}{\lambda_1} E[W_n^{-\ell}] + \frac{1}{\beta_2(1+c)} E[W_n^{1+c}]}.$$

2.3 Threshold Conserving Slow-start

In this section we will consider algorithms that don't change the threshold window size when a packet loss is detected during slow start. However, the algorithms we discuss here will respond to a packet loss detected during slow-start by resetting the window size to zero. Hence the behavior of these protocol differs from the those of the previous section during the slow-start period, see Figure 3. As in the previous section, let S_n denote the instances when the protocol starts a congestion avoidance mode, and let $W_n = W(S_n)$. Given W_n, W_{n+1} is determined only by the time until the first packet loss detection after S_n. Hence the $\{W_n\}$ form a Markov process with the same transition probabilities as in Section 2.1, and hence with the same stationary probability distribution.

Note now that the interval between successive starts of a congestion avoidance phase consists of the following: a period where the process is in congestion avoidance mode, zero or more periods during slow-start mode in which the process is reset to zero before it reaches the threshold, and a final period during slow-start mode in which the process reaches the threshold and subsequently enters the next congestion avoidance phase. The last of these periods is the same as the slow-start period of the previous section. The others correspond to slow-start periods where a packet loss is detected before the threshold is reached. Since each of these periods start with the window size equal to zero, the times spent in each form an independent identically distributed sequence of random variables. (See Figure 3.) Conditioned on W_{n+1}, the number of such periods is a geometric random variable.

With these observations it is possible to extend the results in the previous section to this protocol. The reader is referred to [3] for the details.



The figure shows part of the sample path of the threshold conserving slow-start process. We see that after the congestion avoidance period $T_{0,ca}$ the process enters slow-start, during which three packet losses occur, and the window size is reset to zero at each. ($N_0 = 3$.) The threshold (shown by the dotted line) is reached only at the fourth try, and the process re-enters the congestion avoidance phase.

Figure 3: A sample path of the window size for threshold conserving slow start.

3 Conclusion

We have used stochastic differential equations to model a broad class of window protocols. Our models also allow for the possibility that the packet loss rate depends on the state of the protocol through the window size. Our analysis gives information on the distribution of the window size not just at a random point but at points in time that are relevant to the protocol, such as the detection of a packet loss. The protocols that can be analyzed by our method allow the possibility of a slow-start mode in addition to a congestion avoidance mode.

Acknowledgements

We thank David Lee for bringing this problem to our attention and very helpful discussions. We would also like to thank S. Borst for informing us of [2], and to J. E. Mazo for careful comments.

References

[1] T. V. Lakshman, U. Madhow, "The Performance of TCP/IP for Networks with High Delay-Bandwidth Products and Random Loss," *IEEE/ACM Transactions on Networking*, vol. 5, No. 3, pp. 336–350, June, 1997.

[2] T. J. Ott, J. H. B. Kemperman, M. Mathis, "The Stationary Behavior of Ideal TCP Congestion Avoidance," preprint, August 1996. `ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps`

[3] S. A. Savari and İ. E. Telatar, "The Behavior of Certain Stochastic Processes Arising in Window Protocols," *Submitted to IEEE Trans. on Inf. Theory*, 1998.