



Comparative Study of Different Adaptive Window Protocols

A. A. KHERANI*

B. J. PRABHU

K. E. AVRACHENKOV

E. ALTMAN

INRIA, 2004 Route des Lucioles, Sophia-Antipolis, France

alam@cse.iitd.ac.in

Ext-Balakrishna.Prabhu@vtt.fi

K.Avrachenkov@sophia.inria.fr

Eitan.Altman@sophia.inria.fr

Abstract. We study an Adaptive Window Protocol (AWP) with general increase and decrease profiles in the presence of window dependent random losses. We derive a steady-state Kolmogorov equation, and then obtain its solution in analytic form for particular TCP versions proposed for high speed networks, such as Scalable TCP and HighSpeed TCP. We also relate window evolution under an AWP to workload process in queueing systems; this observation gives us a way to compare various AWP protocols.

Keywords: transmission control protocol (TCP), adaptive window protocol (AWP), steady-state Kolmogorov equation, scalable TCP, HighSpeed TCP

1. Introduction

Over the years, the transmission control protocol (TCP) [9] has satisfactorily handled the majority of reliable data transfers in the Internet. The TCP algorithm probes the network for the bandwidth that it can obtain. The sender transmits a bunch (also called as window) of data packets, which, if successful, are acknowledged by the receiver. The sender interprets the successful reception of packets as a sign of available bandwidth, and reacts by transmitting packets at a higher data rate. When the total input rate to the network exceeds the capacity, the network reacts by dropping some of the packets. The unsuccessful transmission of packets causes the sender to reduce its transmission rate. This simple but robust algorithm has performed quite well in networks with low bandwidth. However, the Internet itself has grown and evolved during this time. The present versions of TCP were designed when the available bandwidth in the Internet was significantly smaller than the available bandwidth today. The low available bandwidth led to a window increase algorithm which was conservative and not fast enough to make efficient use of the large available bandwidth.

The inability of the present versions of TCP to rapidly attain high transmission rates has resulted in several proposed modifications—examples include HighSpeed TCP [8], Scalable TCP [10], Westwood+, CuBIC, and FAST. Data transfer protocols operating

*Corresponding author.

Present address: Computer Science Department at Indian Institute of Technology, Delhi, India

in these networks are expected to maintain a very high window size (i.e., a high data transmission rate). Most of the proposals therefore suggest a window increase algorithm which is faster and a decrease algorithm which is less conservative than the present TCP. Comparative study of such protocols is an important issue. The analytical models can give insights into the behaviour of these protocols. The study could also include, for example, conditions under which two AWP's behave similarly. Since it is always desirable to use a protocol which is easier to implement and does not have many parameters to be tuned, such a study will provide some guidelines on the TCP version to employ.

The TCP versions (presently deployed, and the new proposals) can be coarsely classified in two categories. The classification is based on signals which are interpreted as congestion. The "loss-based" algorithms interpret only packet losses as signs of congestion. Examples of TCP versions in this category are Tahoe, New Reno, SACK (which are presently deployed), Scalable, and HighSpeed TCP (both of which proposed enhancements). In addition to packet losses, the "delay-based" algorithms also use variations in the round trip delay to estimate the available bandwidth. TCP Vegas, FAST and Westwood+ are examples of "delay-based" protocols.

In this paper, we aim to study analytically the behaviour of a class of "loss-based" algorithms. The "loss-based" algorithms can be described by the increase algorithm (when there are no packet losses) and the decrease algorithm (which is in response to a packet loss). Therefore, the building blocks for modelling a "loss-based" algorithm are its increase and decrease algorithms, and the packet loss characteristics in the network. The packet losses in the network are frequently modelled as independent of the current transmission rate. However, at high transmission rates the probability of a packet loss due to link layer errors (for example, errors due to imperfections in the fiber optic cables) is high. Since TCP is unable to distinguish between the various causes of packet loss and interprets every loss/drop as a sign of congestion, it becomes necessary to consider a packet loss probability which depends on the present transmission rate. Another reason for considering window dependent loss rate is the following. The loss process seen by a TCP sender may have its origin in deliberate marking/dropping owing to some active queue management (AQM) scheme employed in the network, in congestion losses, or in link errors. The rate of receiving such a signal will depend on the window process itself (see [7] for related discussion). Hence in our study we consider a general state (window) dependent loss rate.

1.1. Analytical models for "loss-based" TCP algorithms

In this section we briefly describe the various models that have been used to study the performance of the TCP algorithm. A detailed literature review is also presented in [7].

Modelling and analyzing a large network with multiple flows is quite involved. The first efforts in modelling of TCP were directed towards a single connection with a large amount of data to send [14]. The connection was subject to independent packet losses. A fluid approximation of the discrete window process was introduced. This resulted in an elegant "inverse square-root" relation between the throughput and the packet loss

probability. The distribution of the fluid approximation was also provided. The model did not consider neither time-outs nor a receiver limitation on the sender's window size. However, the stochastic modelling of the loss process and the fluid approximation of the window size process became the basis for other models which incorporated the time-outs and receiver limitation (see [7] and references therein). We follow similar modelling approach for a general loss-based protocol with state (window) dependent loss probabilities.

There is a vast amount of literature on TCP modeling, and any attempt to cite even a moderate part of it would be lengthy enough. For an extensive literature survey on TCP modeling the reader is referred to [7].

The contribution (and organization) of this paper is as follows.

- In Section 2, we give a characterization of a general AWP, and identify the various quantities that determine the performance of such protocols. Kolmogorov equations for the stationary probability measure are then derived.
- In Section 3, we give conditions under which two AWP's have related stationary distribution. Furthermore, we demonstrate that the window process under a multiplicative decrease protocol is also related to the workload process in a queueing system with workload dependent service and arrival rates.
- In Sections 4 and 5, we solve the Kolmogorov equations to study the performance of recently proposed TCP modifications (Scalable TCP [10] and HighSpeed TCP [8]). In these sections, we consider two different forms of loss rates: constant and linear. The analysis also provides insights into the sensitivity of system performance to the parameters of the AWP employed.
- In Section 6, we compare simulation results with the results of Sections 3–5.

2. The model

We consider an AWP controlled persistent file transfer over an Internet (bottleneck) link. For applications using HighSpeed and Scalable TCP this link will typically be a very high bandwidth delay product link. We assume that the connection is long enough to see a stationary regime and that its throughput performance is governed by the steady state regime (see [2] for justification of this assumption). Applications using HighSpeed TCP and Scalable TCP typically transfer very large volume files. Therefore, studying persistent transfers is justified and important in such cases. We model the loss process as a Poisson process with a time varying intensity that depends on the instantaneous window size of the AWP [7]. These losses could be owing to congestion losses, random link losses or some deliberate packet marking/dropping by the router buffer using an AQM. As is common in related studies [2,4,11], we consider the evolution of window as an infinitely divisible fluid. Details of the model are given below.

Let x_t denote the window size of the AWP at time instant t (note that we are not specifying the initial window size x_0 here, thus assuming a stationary window process).

We now give the description of the window evolution. In the absence of losses, the window increase in time interval $[t, t + \Delta]$ is given by

$$x_{t+\Delta} = x_t + f(x_t)\Delta + o(\Delta). \quad (1)$$

where $f(\cdot)$ is a function bounded below by some positive quantity. We also assume that there is a lower bound on the window size, denoted by x_{\min} .

The increase in window cannot continue forever because drops owing to congestion or channel losses or AQM marking can occur at random instants in time.¹ Let $N(t)$ be the counting process corresponding to the loss events, i.e. $N(t) - N(t - u)$ is the number of losses in time interval $(t - u, t)$. In what follows, we assume that $N(t)$ is a Poisson process with time varying intensity. Further, we assume that the instantaneous rate of the $N(t)$ process depends only on the current window size, x_t , of the connection. Let $\lambda(x)$ be the rate of the $N(t)$ process when the window size, x_t , is x . Each loss results in a window reduction (this is because TCP assumes that each packet drop/mark corresponds to a congestion event in the network). Under the fluid model, it is standard to assume that this window reduction is reflected as an instantaneous jump in the x_t process. The assumptions imply that $P\{N(t + \Delta) - N(t) = 1\} = 1 - P\{N(t + \Delta) - N(t) = 0\} = \lambda(x_t)\Delta + o(\Delta)$. Thus, for small Δ , if $N(t + \Delta) - N(t) = 1$, the window is instantaneously reduced as

$$x_{t+\Delta} = G(x_t) + o(\Delta), \quad (2)$$

for some continuous function $G(\cdot)$ such that $G(x) < x$ and $G(x_{\min}) = x_{\min}$. We assume that $G(\cdot)$ is such that if $x_1 < x_2$ then either $G(x_1) < G(x_2)$ or $G(x_1) = G(x_2) = x_{\min}$. The assumption of continuity on $G(\cdot)$ implies that the set $s(x) = \{u \geq x : G(u) \leq x\}$ is connected. Define also $H(x) = \sup\{u \geq x : G(u) \leq x\} = \sup s(x)$; we will also use the notation $G^{-1}(x)$ to mean $H(x)$. Note here that, unlike [2], we are assuming that $G(\cdot)$ is a deterministic function. This is true in new versions of TCP which decrease the window at most once in a round trip time. Similar modeling assumption for decrease is also made in [4,7]. The above continuous-time evolution model can be obtained from a discrete-time evolution using the approach of [7]. For convenience, the approach is outlined in Appendix.

2.1. Incorporating a bound on the window size

The window evolution process described above does not incorporate any bound on the maximum allowed window size. In practice, however, there will be an upper bound M on the window size that the AWP is allowed to use. This bound usually is either the receiver's advertised window (which is the maximum number of packets that the

¹ Congestion losses occur also when the window size reaches the practical limit of the total round trip pipe size (sum of the link bandwidth-delay product and the router buffer). This aspect of congestion losses will be addressed later in this section. For presentation of the basic model, we assume here that there is no upper bound on the values that the window can take.

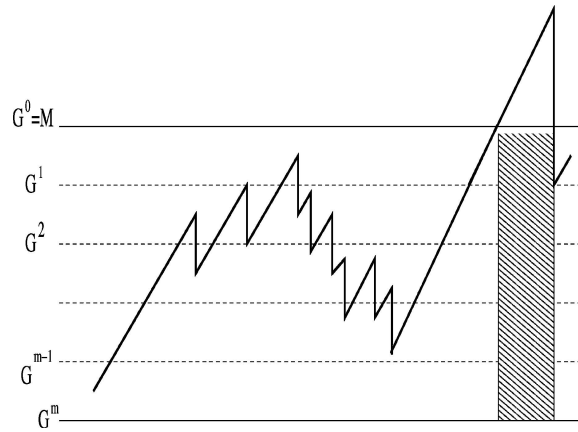


Figure 1. Evolution of the transformed window process $\{x_t\}$.

receiving entity's receive buffer can accommodate) or the total round trip pipe size. The behavior of the AWP under these two bounds is very different. In the first case where the window is restricted by the receiver's advertised window M , the window size stays at this value until a loss event takes place. While in the second case where M represents the round trip pipe size, reaching this limit results in an instantaneous congestion loss and the window size is reduced. However, since the loss rate is assumed to be function of window size alone, it follows that we can study the second case via the first case (for details, see [2] which also addresses this issue for a constant loss rate). Hence in what follows we will restrict ourselves to the case where M represents the window limitation owing to the receiver's advertised window.

Assume that the range of the values of the window process is divided into the intervals between points $[H^j(x_{\min}), H^{j+1}(x_{\min})]$ where H^j is j -fold composition of $H(\cdot)$ with itself and let $H^0(x_{\min}) \triangleq x_{\min}$. Consider an M such that $M = H^m(x_{\min})$ for some $m \geq 1$. Note that, under our choice of M , $H^j(x_{\min}) = G^{m-j}(M)$ with $G^0 \triangleq M$ and $G^i = G(G^{i-1})$. Under the above definitions, $x \in [G^i, G^{i-1}] \Rightarrow H(x) \in [G^{i-1}, G^{i-2}]$. The case where such an m does not exist, i.e., $H^{m-1} < M < H^m$ for some m , is not possible since the definition of $G(\cdot)$ depends on x_{\min} and M implicitly, and it ensures that $G(G^{m-1}) = x_{\min}$ so that $H^m = M$.

We consider a further modification in the evolution of the window process $\{x_t\}$; this is shown in figure 1. For this modified process, the window size is unbounded. However, when $x > G^0$, we assume that the loss rate is constant and equal to $\lambda(G^0)$ and that the window increase is linear, i.e., $f(x) = 1$ for $x > G^0$. We also assume that if a loss event takes place when $x \geq G^0$, the window is dropped to $G^1 = G(M) = H^{m-1}(x_{\min})$. The evolution of the modified process for $x < G^0$ is unchanged, i.e., a loss event occurs with rate $\lambda(x_t)$ and the window is dropped to $G(x_t)$ in case of a loss event when $x_t < G^0$. Thus, the modified process has the following evolution: the increase profile is given by

$$x_{t+\Delta} = x_t + \Delta f(x_t) + o(\Delta).$$

Losses occur according to a Poisson process of rate $\lambda(x_t \wedge G^0)$ and the window reduction in case of a loss event in time interval $(t, t + \Delta)$ is

$$x_{t+\Delta} = G(x_t \wedge G^0).$$

Remark. If the window size is bounded (as the case will be in the rest of this paper), so is $\lambda(\cdot)$. In this scenario, it is sometimes convenient to assume that the process $N(t)$ is actually derived from a standard Poisson process $\Lambda(t)$ of unit rate so that a jump in $\Lambda(t)$ results in a jump in $N(t)$ only with probability $\frac{\lambda(x_t)}{\sup_x \lambda(x)}$.

2.2. Performance measure

There can be various performance measures of interest in the context of the problem under consideration. Most prominent of these (and the one most frequently used in literature) is the expected window size. However, finding only expected window size may not give much information about the window process itself. An analysis for the performance of an AWP should also consider the stationary window size distribution. In this paper we are interested in obtaining the stationary window size distribution of the AWP.

2.3. The Kolmogorov equations

Let $\pi(x)$ be the density function and $\Pi(x)$ be the distribution function of the (modified) x_t process (note that we are suppressing the dependence on M here.).

Let, for a fixed t , $[t, t + \Delta]$ be a small time interval. When the process is in equilibrium, the probability of upcrossing level x during $[t, t + \Delta]$ is

$$\begin{aligned} P\{x_t \in (x - \Delta f(x), x)\} P\{\text{no loss during } [t, t + \Delta]\} \\ = \pi(x) \Delta f(x) (1 - \lambda(x) \Delta) + o(\Delta) \end{aligned}$$

Similarly, the probability of downcrossings is

$$\begin{aligned} \int_{u=x}^{\infty} P\{x_t \in (u, u + du)\} \lambda(G^0) \Delta &= \int_{u=\bar{x}}^{\infty} \pi(u) \lambda(G^0) du \Delta + o(\Delta) \quad x \geq G^0, \\ \int_{u=x}^{\infty} P\{x_t \in (u, u + du)\} \lambda(u \wedge G^0) \Delta &= \int_{u=x}^{G^0} \pi(u) \lambda(u \wedge G^0) du \Delta + o(\Delta) \\ &\quad G^1 < x \leq G^0, \\ \int_{u=x}^{H(x)} P\{x_t \in (u, u + du)\} \lambda(u) \Delta &= \int_{u=x}^{H(x)} \pi(u) \lambda(u) du \Delta + o(\Delta) \\ &\quad x_{\min} \leq x < G^1. \end{aligned}$$

In the steady state, the probability of up-crossing is equal to that of down-crossing. Thus,

letting $\Delta \rightarrow 0$, we obtain

$$f(x)\pi(x) = \begin{cases} \int_{u=G^0}^{\infty} \pi(u)\lambda(G^0)du = \lambda(G^0)\Pi^c(x), & x \geq G^0, \\ \int_{u=G^0}^{u=x} \pi(u)\lambda(u)du + \lambda(G^0)\Pi^c(G^0), & G^1 < x \leq G^0, \\ \int_{u=x}^{u=\bar{H}(x)} \pi(u)\lambda(u)du, & x_{\min} \leq x < G^1. \end{cases}$$

Using integrating factor method for the Kolmogorov equation for $x \geq G^0$,

$$\Pi^c(x) = \Pi^c(G^0)e^{-\lambda(G^0)(x-G^0)}, \quad x \geq G^0.$$

The basic idea involved in solving the Kolmogorov equations obtained above is to use the knowledge of $\lambda(\cdot)$ to obtain $\pi(x)$ for $x \in [G^1, G^0]$. Now, one can find $\pi(x)$ for $x \in [G^{i+1}, G^i]$ from the knowledge of $\pi(x)$ for $x \in [G^i, G^{i-1}]$. In this process, since we need to integrate over different regions, integration constants appear naturally. These integration constants are computed using continuity of $\Pi(\cdot)$ at the boundaries G^i . Clearly, the form of $\pi(\cdot)$ will depend on that of $\lambda(\cdot)$ and of $H(\cdot)$.

In this paper we will be working with a bounded window process, and when we write Kolmogorov equations for different protocols, we will not give the detailed equations as done above. We will ignore the boundary conditions at the upper and lower bounds for sake of presentation clarity.

3. Relations between two systems of window evolution

We now consider two systems, 1 and 2, having their own increase profile, decrease profile and loss rates denoted by $f_i(\cdot)$, $G_i(\cdot)$ and $\lambda_i(\cdot)$, respectively, $i \in \{1, 2\}$. We provide a condition under which these two systems have related stationary probability distribution. Assuming that $G_1(x) = G_2(x) = G(x)$, $\forall x$, and that in both the systems the upper bound on the window is the same (and is equal to M), the Kolmogorov equations for the two systems are²

$$f_i(x)\pi_i(x) = \int_{u=x}^{G^{-1}(x)} \lambda_i(u)\pi_i(u) du ,$$

or, equivalently,

$$\frac{f_i(x)\lambda_i(x)\pi_i(x)}{\lambda_i(x)E[\lambda_i(X)]} = \int_{u=x}^{G^{-1}(x)} \frac{\lambda_i(u)\pi_i(u)}{E[\lambda_i(X)]} du$$

²It should be noted here that the boundary considerations are automatically taken care of by using the convention that for $x \in [G^1, G^0]$, we can take $G^{-1}(x) = \infty$. So that the kolmogorov equations for any value of x is given by the mentioned single equation. Here one more convention is followed. i.e., $\lambda(x) = \lambda(G^0)$ for $x \geq G^0$.

where $E[\lambda_i(X)] = \int_x \lambda_i(x)\pi_i(x)dx$ is the expected loss rate in the i th system. It is clear from the above set of equations that if $\frac{f_1(x)}{\lambda_1(x)} = \frac{f_2(x)}{\lambda_2(x)}, \forall x$, the functions $\frac{\lambda_1(x)\pi_1(x)}{E[\lambda_1(X)]}$ and $\frac{\lambda_2(x)\pi_2(x)}{E[\lambda_2(X)]}$, both being probability density functions integrating to unity, are equal for each x . Thus, we have

Theorem 1. If two AWP controlled window evolutions are such that both have same drop profile and both have the same ratio of increase profile to the loss rate for all x , then

$$\frac{\pi_1(x)}{\pi_2(x)} = C \frac{\lambda_2(x)}{\lambda_1(x)} = C \frac{f_2(x)}{f_1(x)},$$

where $C = \frac{E[\lambda_1(X)]}{E[\lambda_2(X)]}$.

This result is important as it gives us a way to analyse one system using the analysis of the other related system. We use this result in Section 4.2 where we use the observation that an AIMD protocol with constant loss rate and an MIMD protocol with linear loss rate satisfy the requirement of Theorem 1 as for the first (AIMD) system $f(x) = \alpha$ and $\lambda(x) = \lambda$ while for the second (MIMD) system $f(x) = \alpha x$ and $\lambda(x) = \lambda x$ and both have same multiplicative decrease factor. Since the analysis for the first system is known from [2], we use it to find stationary distribution for the MIMD protocol with linear loss rate.

In the special case where both the system use multiplicative decrease profile with a constant decrease factor β , we can get some more detailed equivalence between two related systems. This is done next.

3.1. A queueing model for multiplicative decrease protocols

Consider an AWP with a constant multiplicative decrease factor β . Introduce the transformation $z_t = \ln M - \ln x_t$. We are assuming that z_t is unbounded, i.e., that $x_{\min} = 0$; we can do this since we can use standard approach ([3], Chapter 14) to analyse the case where z_t is bounded by $\ln M - \ln x_{\min}$ from that where z_t is unbounded. The evolution of the process z_t now is as shown in Figure 2. It is evident from the transformation (as also visualised in the figure), the multiplicative decrease of the process x_t presents itself as a *constant* increase of $\ln \beta$ in the evolution of process z_t . The evolution of process z_t suggests that z_t can be thought of as workload process of a queue for which the service requirement of the customers is constant ($-\ln \beta$). If the increase profile and loss rate for x_t process are $f(\cdot)$ and $\lambda(\cdot)$, then in the z_t process, the customer arrival rate is $\lambda(Me^{-z_t})$ and service rate is $\frac{f(Me^{-z_t})}{Me^{-z_t}}$, both depending on the workload process z_t . Thus we get a queueing system with constant service requirements and state dependent service rates and arrival rates. This observation leads us to the following theorem.

Theorem 2. Consider window evolutions in the two systems 1 and 2 introduced above, both with same multiplicative decrease profile. If $\frac{f_1(x)}{\lambda_1(x)} = \frac{f_2(x)}{\lambda_2(x)}$ then the distribution of window size just before loss instants is *same* in both the systems.

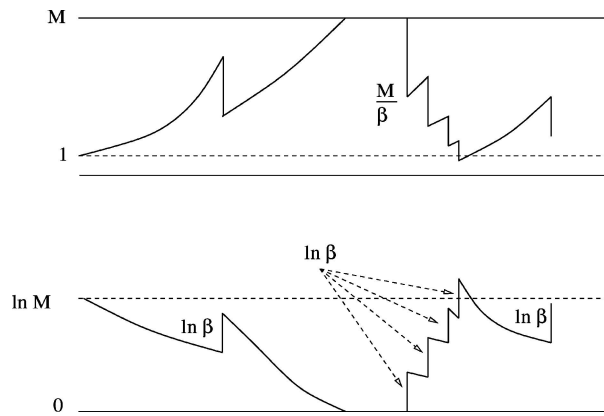


Figure 2. The original window evolution (top) and its transformation to the workload process in a queue (bottom).

Proof. The logarithmic transformation introduced above maps the two system into queueing systems with constant service requirements. The proof then follows from ([6], Theorem 3.3) which says that for two queueing systems with same service requirement distribution if the ratio of the two arrival rates is same as that of their service rates for any workload, then the stationary distribution of the workload process seen just before an arrival is same for both the system. The proof follows from the relation between the loss rate in window process and the arrival rate in the queueing system and that between the increase profile in the window process and the service rate in the queueing system. \square

Applying this result to the two systems satisfying the above condition where the first one is AIMD with constant loss rate and the second one is MIMD with linear loss rate, we see that the stationary distribution of the window process just before (and hence just after) loss instants is same. Thus, the standard AIMD protocol with constant loss rate is same as MIMD protocol with linear loss rate in the sense that the distribution of the window sizes just before losses are the *same* for the two.

Further, since Theorem 2 is valid for any two AWP's satisfying the required conditions, it is seen that if for one of the AWP's the loss rate is constant, the PASTA property implies that the stationary (time average) distribution of the window size in the system with constant loss rate is same as the window size distribution just before losses in either of the system.

Theorem 3. Consider window evolutions in the two systems 1 and 2 introduced above, both with same multiplicative decrease profile. If $\frac{f_1(x)}{\lambda_1(x)} = \frac{f_2(x)}{\lambda_2(x)}$ then the time average

distribution of window size $\pi_i(\cdot)$ in the two systems is related by

$$\frac{\pi_1(x)}{\pi_2(x)} = C \frac{f_2(x)}{f_1(x)} = C \frac{\lambda_2(x)}{\lambda_1(x)}$$

where $C = \frac{\lambda_1(M)\Pi_1^c(M)}{\lambda_2(M)\Pi_2^c(M)}$ with $\Pi_i^c(\cdot)$ denoting the complementary distribution function.

Proof. Follows from ([6], Theorem 3.1) which states that the two queueing systems if the ratio of the two arrival rates is same as that of their service rates for any workload, then the density corresponding to the time average stationary distribution of the two systems $\pi_1(z)$ and $\pi_2(z)$ are such that $\pi_1(z)r_2(z) = C\pi_2(z)r_1(z)$ where $r_i(\cdot)$ is the service rate in the i th system and $C = \frac{\Pi_1(0)}{\Pi_2(0)}$. The proof follows from the relation between the loss rate in window process and the arrival rate in the queueing system and that between the increase profile in the window process and the service rate in the queueing system. \square

Remark. It is important to note that the window process with a lower bound of 1 and an upper bound of $M < \infty$ is always ergodic in the case of multiplicative decrease algorithm. This is because for any bounded loss rate and positive increase profile, the window process $\{x_t\}$ is irreducible. However, if we assume $x_{\min} = 0$, then the corresponding unbounded transformed queueing process need not always be ergodic. Thus, we can not always use the truncation method of [3] mentioned above. Hence it becomes necessary to solve the detailed Kolmogorov equations for each case. This remark is, in particular, relevant for the case where the AWP follows a multiplicative increase multiplicative decrease algorithm and the loss rate is constant. For this case the transformed process z_t is just the workload process of an M/D/I queue. However we can not use this approach for $\lambda > -\ln \beta$ owing to the above mentioned reason.

Remark. The process $z_t \triangleq M - x_t$ always represents the workload process in a queue with state dependent arrival rate, service rate and service requirement.

Remark. The results of this section indicate that if the losses come from an AQM scheme, then there are many AWP-AQM pairs (i.e., $f(\cdot)$ and $\lambda(\cdot)$) which have the same drop profile ($G(\cdot)$) and have similar performance (in the sense of Theorem 1). Moreover, if the decrease profile is fixed to be a multiplicative one, we see that all these AWP-AQM pairs have *same* window distribution before drop instants (Theorem 2).

Having made the relation between the evolution of the window process of the AWP and the workload process in a queueing system, we now proceed to solve the Kolmogorov equations considering specific forms of the $\lambda(\cdot)$ and $H(\cdot)$ functions. As remarked above, analysing the queueing system does not provide us with the stationary distribution for all the possible values of the involved parameters. This makes it necessary to solve the Kolmogorov equations for each instance of the problem.

Till now the development did not consider exact form of loss rate $\lambda(\cdot)$ and AWP. In the following sections we consider specific forms of $\lambda(\cdot)$ to find the stationary window

size distribution and work out the solution of Kolmogorov equation for several available TCP versions. We first analyse, under constant and linear loss rate, Scalable TCP [10] which represents a class of MIMD protocols; this is done in Section 4. We then consider the situation of constant and linearly increasing loss rates for HighSpeed TCP [8] in Section 5.

We remark here that a linear loss rate, $\lambda(x) = \lambda x$ is suitable for the cases where, like NewReno version of TCP, only one window reduction takes place irrespective of the number of losses in a round-trip time and each packet is dropped with a fixed probability. Since high speed networks are expected to have most of the losses coming from link layer losses, assuming a fixed packet drop probability (which may also include congestion losses) is reasonable, and important.

4. MIMD protocols with bounded window (Scalable TCP)

For the case of MIMD protocols. Scalable TCP being an important example of such protocols, the window evolution is described as follows. In case of no loss in interval $[t, t + \Delta]$, the window increases to

$$x_{t+\Delta} = (x_t + \alpha x_t \Delta + o(\Delta)) \wedge M, \quad (3)$$

for some $\alpha > 0$ and an upper bound M on the window size. In case of a loss in interval $[t, t + \Delta]$, the window decreases to

$$x_{t+\Delta} = (\beta x_t) \vee 1 + o(\Delta),$$

where $1 > \beta > 0$ is the multiplicative decrease constant. The natural lower bound of $x_t \geq 1$ packet applies.

4.1. Constant loss rate

It is clear now that the transformation $x_t \mapsto \frac{\log x_t}{\alpha} \triangleq y_t$ results in the process $\{y_t\}$ having linear increase profile. The transformed window after a loss event in interval $[t, t + \Delta]$ is given by

$$y_{t+\Delta} = (y_t - \theta)^+ + o(\Delta),$$

where $\theta \triangleq \frac{-\log(\beta)}{\alpha} > 0$. The transformed process $\{y_t\}$ can in itself be viewed as window evolution under another AWP for which, $y_{\min} = 0$ and $H(y) = y + \theta$. Clearly, one can obtain the stationary distribution for the original process $\{x_t\}$ from that of $\{y_t\}$, hence it is enough to solve for the stationary distribution of the process $\{y_t\}$ in order to get that for the process $\{x_t\}$. Hence, in the rest of this subsection we will work only with the process $\{y_t\}$. As will be seen next, the Kolmogorov equations for this process have some special structure which makes it easier to solve.

Now, from the construction of the (virtually) bounded process of Section 2, $G^0 = M = m\theta$ and $G^l = (m-l)\theta$, $G(u) = (u-\theta)^+$. This system has simple expressions for up and down crossing rates for $0 \leq y \leq (m-1)\theta$,

$$\pi(y) = \lambda \int_{u=y}^{y+\theta} \pi(u) du,$$

$$\Pi'(y) = \lambda(\Pi(y+\theta) - \Pi(y)),$$

where $\pi(\cdot)$ and $\Pi(\cdot)$ are the density and distribution functions respectively for the process $\{y_t\}$. Defining, for convenience $\Pi_k(y) = \Pi(k\theta + y)$ for $0 \leq y \leq \theta$, we have

Proposition 1. For $0 \leq k \leq m-1$ and $0 \leq x \leq \theta$,

$$\Pi_k^c(x) = \sum_{j=0}^{m-k-1} \Pi_{k+j}^c(0) \frac{(\lambda x)^j}{j!}.$$

Proof. See Appendix. □

Proposition 2. The constants $\Pi_k^c(0)$ are given by

$$\Pi_{m-1}^c(0) = [(a^{m-1} - \phi_1(m-1)) + \sum_{s=1}^{m-3} (-1)^s \sum_{l=s}^{m-2} \phi_s(l)(a^{m-l-1} - \phi_1(m-l-1)) + (-1)^{m-2}(a-b)\phi_{m-2}(m-2)]^{-1}$$

and for $0 \leq k \leq m-2$,

$$\Pi_k^c(0) = \Pi_{m-1}^c(0)[(a^{m-k-1} - \phi_1(m-k-1)) + (-1)^{m-k-2}(a-b)\phi_{m-k-2}(m-k-2) + \sum_{s=1}^{m-k-3} (-1)^s \sum_{l=s}^{m-k-2} \phi_s(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))].$$

with $a = e^{\lambda\theta}$ and $\phi_j(l)$ defined recursively as, $\phi_0(0) = 0$ and

$$\phi_{j+1}(l) = \sum_{s=1}^{l-j} \phi_1(s)\phi_j(l-s), \quad j \geq 1.$$

Proof. See Appendix. □

Remark. For the above case where $x_{\min} = 1$, the evolution of process $\frac{\log M}{\alpha} - \frac{\log x_t}{\alpha} = \frac{\log M}{\alpha} - y_t$ corresponds to the workload process of an M/D/1 queue with a bounded workload capacity of $\frac{\log M}{\alpha}$ and service requirement of θ for each customer. This is a system similar to that of [15] with a difference that the model of [15] assumes that the customer that can make the workload to exceed a certain fixed threshold is lost. While in

our case such a customer is not completely lost but is admitted with a service that makes the workload process equal to the threshold. Thus the above result is of independent interest in queueing theory.

Remark. We can also easily incorporate another value of $0 < x_{\min} \neq 1$ in the above analysis. As mentioned in Section 3.1, if we assume that $x_{\min} = 0$, the transformation $\frac{\log M}{\alpha} - \frac{\log x_t}{\alpha}$ corresponds to the workload process of a classical M/D/1 queue. For this case the moments and the stationary window size distribution are well known.

(1) *MIMD with Unbounded Window: A D/M/1 Queue:* Assuming that $M = \infty$. i.e., there is no bound on the window size, we can not use the results from above directly in this case. Another approach to obtain the stationary distribution $\Pi(\cdot)$ is to look at the process $\{y_n, n \geq 0\}$ embedded just after the loss instants of the transformed process with linear increase profile, $\{y_t\}$. Let $\{a_n, n \geq 0\}$ denote the time between two successive losses. Then, $\{y_n\}$ is a continuous state space Markov chain which is given by the recursive equation

$$y_{n+1} = (y_n + a_n - \theta)^+. \tag{4}$$

We note that the loss process a_n is exponentially distributed with rate λ . Equation (4) is the same as the recursive equation for the workload in a D/M/1 queue with interarrival time θ and mean service time $\frac{1}{\lambda}$. The steady state distribution of y , $P(y_n \leq y)$ can be obtained as [12]

$$P(y_n > y) = \left(1 - \frac{s_1}{\lambda}\right) e^{-s_1 y}, \tag{5}$$

where s_1 is the root of the equation $s + \lambda = \lambda e^{s/\theta}$ in $Re(s) < 0$. The stability condition for the workload process of this D/M/1 queue (and, equivalently, for the window size process $\{y_t\}$) is $\theta > \frac{1}{\lambda}$.

In order to obtain the distribution at a random arrival instant, we note that the window size just before loss instant, y_{n+1}^- , is given by

$$y_{n+1}^- = y_n + a_n.$$

Since a_n 's are exponentially distributed with parameter λ ,

$$\begin{aligned} P(y_{n+1}^- > y) &= \lambda \int_0^\infty P(y_n > y - a) e^{-\lambda a} da \\ &= \lambda \int_y^\infty e^{-\lambda a} da + \lambda \int_0^y P(y_n > y - a) e^{-\lambda a} da \\ &= e^{-\lambda y} + \lambda \left(1 - \frac{s_1}{\lambda}\right) e^{-s_1 y} \int_0^y e^{-(\lambda - s_1)a} da \\ &= e^{-s_1 y}. \end{aligned}$$

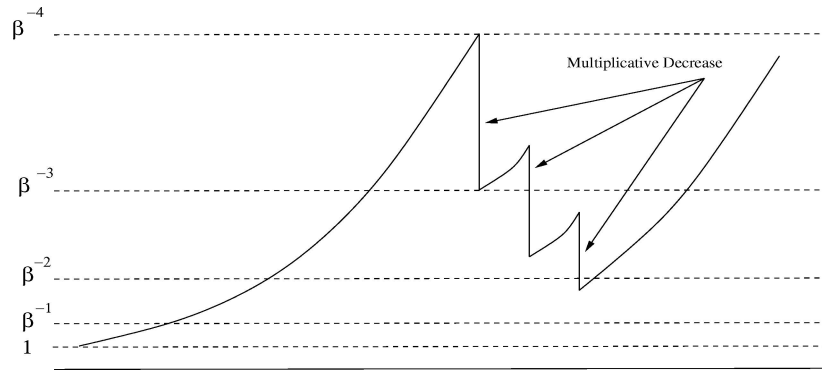


Figure 3. Window evolution under MIMD protocol like Scalable TCP with a lower bound on window size.

Using PASTA property, the window size distribution at a random time is the same as that seen by the loss arrivals. Since $y = \frac{\log x}{\alpha}$, the window distribution at any random time is

$$P(x_t > x) = x^{-\frac{1}{\alpha}}. \quad (6)$$

Remark. This approach can also be used for bounded window process when loss rate is large enough so that the bound is attained with negligible probability.

4.2. MIMD protocols with linear loss rates

For the case of MIMD protocols, the window evolution is described as follows. In case of no loss in interval $[t, t + \Delta]$, the window increases to (assuming no upper bound on window size)

$$x_{t+\Delta} = x_t + \alpha x_t \Delta + o(\Delta), \quad (7)$$

for some $\alpha > 0$. In case of a loss in interval $[t, t + \Delta]$, the window decreases to

$$x_{t+\Delta} = (\beta x_t) \vee 1 + o(\Delta),$$

where $1 > \beta > 0$ is the multiplicative decrease constant. The natural lower bound of $x_t \geq 1$ packet applies.

The window is bounded below by a constraint of x_{\min} packet. The window evolution under such scenario is depicted in figure 3. The figure shows that the window starts evolving from an initial value of 1 packet. There are some multiplicative decrease of window owing to random losses. The vertical axis is shown to be divided into various intervals $I_k \Delta(\beta^{-k}, \beta^{-k-1}]$. Here $\beta < 1$ is the multiplicative decrease factor. The choice of these intervals is explained by the fact that if a loss event occurs when the window

size is in interval I_{k+1} then the reduced window is in region I_k . The upper bound on x is $M = \beta^{-m}$ for some m .

For this case the following Kolmogorov equations can be obtained for $x < \beta^{-m+1}$,

$$\pi(x)\alpha x = \int_{u=x}^{u=\frac{x}{\beta}} \pi(u)\lambda u du ,$$

where α is as in Equation (7). Denote now, by an abuse of notation, $\lambda = \frac{\lambda}{\alpha}$. The above Kolmogorov equation is then

$$\pi(x)x = \int_{u=x}^{u=\frac{x}{\beta}} \pi(u)\lambda u du.$$

Proposition 3. The steady state probability density function of the window size under linear loss rate is given by, if $x \in I_{m-k}, k \geq 2$,

$$\pi(x) = M P_M \sum_{i=1}^k c_i^{(k)} \frac{\lambda}{x\beta^{i-1}} e^{\frac{\lambda x}{\beta^{i-1}}}.$$

Here $c_i^{(k)}$ are some constants obtained by normalising $\pi(\cdot)$ to get a probability measure and P_M is the probability mass at M .

Proof. See Appendix for expressions for P_M and $c_i^{(k)}$. □

One is often interested in finding the moments of the window process. This can be easily done without computing the coefficients $c_i^{(k)}$. We assume here that $x_{\min} = 0$ and $M = \infty$; this is expected to approximate the case when the upper and lower bounds are not attained frequently. The Kolmogorov equation obtained above is multiplied by $x^{j-1}, j \geq 0$ to obtain

$$\begin{aligned} \pi(x)x^j &= x^{j-1} \int_{u=x}^{u=\frac{x}{\beta}} \pi(u)\lambda u du \\ \Rightarrow \int_{x=0}^{\infty} \pi(x)x^j dx &= \int_{x=0}^{\infty} x^{j-1} \int_{u=x}^{u=\frac{x}{\beta}} \pi(u)\lambda u du dx \\ E[X]^j &= \int_{u=0}^{\infty} \int_{x=\beta u}^u x^{j-1} dx \pi(u)\lambda u du \\ \Rightarrow E[X] &= \frac{1}{-\lambda \ln(\beta)} \\ E[X]^{j+1} &= \frac{j}{\lambda(1-\beta^j)} E[X]^j = \frac{j!}{\lambda^j \prod_{i=1}^j (1-\beta^i)} E[X], \quad j \geq 1, \end{aligned}$$

thus we get all the moments of the window size distribution. We see from the above that the tail of the window size distribution is exponentially decaying and that all the moments exist.

5. HighSpeed TCP

HighSpeed TCP (HSTCP, [8]) updates the window in a round-trip time according to the following rules: In case of no loss in a round-trip time during which the window size was w , the window is incremented by a window dependent quantity, denoted $a(w)$, so that the new window size is $w + a(w)$, and in case of a packet drop on a round-trip time, the window is decremented by a window dependent factor $b(w)$ so that the new window size is $(1 - b(w))w$. The window size is bounded by two values w_l and w_h and

$$\begin{aligned} a(w) &= \frac{2w^2 b(w) p(w)}{2 - b(w)}, \\ b(w) &= \frac{\log\left(\frac{w}{w_l}\right)}{\log\left(\frac{w_h}{w_l}\right)} (b_h - b_l) + b_l, \\ p(w) &= \exp\left(\frac{\log\left(\frac{w}{w_l}\right)}{\log\left(\frac{w_h}{w_l}\right)} \log\left(\frac{p_h}{p_l}\right) + \log(p_l)\right), \end{aligned}$$

where $b_h = b(w_h)$, $b_l = b(w_l)$, $p_l = p(w_l)$ and $p_h = p(w_h)$ are design parameters.

It is suggested in [8] to set $w_l = 31$ and $p_l = \frac{1.5}{w_l^2}$. Note that

$$p(w) = v w^\mu \quad (8)$$

where

$$\mu = \frac{\log\left(\frac{p_h}{p_l}\right)}{\log\left(\frac{w_h}{w_l}\right)} \quad (9)$$

and

$$\begin{aligned} v &= \frac{p_l}{w_l^\mu}, \quad b(w) = A \log(w) + B \\ A &= \frac{b_h - b_l}{\log\left(\frac{w_h}{w_l}\right)}, \quad B = b_l - A \log(w_l). \end{aligned}$$

Since $b_h < b_l$, $A < 0$ and since $w_h \geq w_l$, $p_h \leq p_l \Rightarrow \mu < 0$. We observe that, if R represents the round-trip time, then

$$w(t + R) = w(t) + a(w(t)) = w(t) + \frac{2w^{2+\mu} b(w) v}{2 - b(w)}. \quad (10)$$

This equation shows the importance of parameter μ in understanding the behavior of HSTCP. For example,

- $\mu = -2$ implies that HSTCP is similar to the standard AIMD algorithm of TCP where in each round-trip time, the window is incremented by a small value (in this case $\frac{2b(w)v}{2-b(w)} \approx vb(w)$).
- $\mu > -2$ gives us a protocol whose window increment increases with the window, for example, taking $\mu = -1$ implies that HSTCP is similar to Scalable TCP in behavior since now the increment is approximately linear in window size.

This observation suggests need for care in tuning the HSTCP parameters. It also implies the possibility of existence of a choice of $\mu \in (-2, -1)$ which is neither as aggressive as Scalable TCP nor conservative as standard TCP. Now we analyse HSTCP assuming that $A \approx 0$ so that the decrease factor is constant. Since the form of function $b(w)$ is a design choice (see [8]), this form of $b(w)$ can be chosen for simplicity of implementation. Further, for this choice of $b(w)$ we can find the stationary window size distribution for the protocol for different values of μ using the following method.

5.1. Constant loss rate

First observe that for $b(w) = B$, the increase profile of the protocol is $f(w) = \frac{2vBw^{2+\mu}}{2-B}$ and assuming a constant loss rate $\lambda(w) = \lambda$, the Kolmogorov equations are

$$\pi(w) \frac{2vBw^{2+\mu}}{2-B} = \int_{u=w}^{\frac{w}{B}} \lambda \pi(u) du$$

which is rewritten as

$$\frac{2vBw^{2+\mu}}{\lambda(2-B)} \pi(w) = \int_{u=w}^{\frac{w}{B}} \pi(u) du. \quad (11)$$

We now introduce a transformation, similar in spirit to what was done in Section 4.1. This requires splitting the possible values of μ into various regions and using separate transforms in each of these:

- $\mu < -1$: Define the process $\{y_t\}$ whose value at instant t is

$$y_t = \frac{\lambda(B-2)}{2vB(1+\mu)w_t^{1+\mu}}$$

where, since $\mu < -1$, the process $\{y_t\}$ is also non-negative. Now, the Kolmogorov equation for the transformed process $\{y_t\}$ is

$$\tilde{\pi}(y) = \int_{u=y}^{\frac{y}{B^{-(1+\mu)}}} \tilde{\pi}(u) du.$$

- $\mu = -1$: corresponds to Scalable TCP, a case we have already studied.
- $\mu > -1$: Define the process $\{y_t\}$ whose value at instant t is

$$y_t = \frac{\lambda(2 - B)}{2\nu B(1 + \mu)w_t^{1+\mu}}$$

where, since $\mu > -1$, the process $\{y_t\}$ is also non-negative. Now, the Kolmogorov equation for the transformed process $\{y_t\}$ is

$$\tilde{\pi}(y) = \int_{u=yB^{(1+\mu)}}^y \tilde{\pi}(u)du.$$

A further transformation of $y \mapsto z = yB^{1+\mu}$ gives another protocol for which the Kolmogorov equation is

$$B^{1+\mu}\hat{\pi}(y) = \int_{u=y}^{yB^{-(1+\mu)}} \hat{\pi}(u)du.$$

The end process thus obtained can be thought of as window evolution under standard AIMD algorithm of TCP, where the increase profile $f(\cdot)$ and the multiplicative decrease factor are

- $f(x) = 1$ and multiplicative decrease factor is $B^{-(1+\mu)} < B < 1$ when $\mu < -1$,
- $f(x) = B^{1+\mu}$ and multiplicative decrease factor is $B^{(1+\mu)} < 1$ when $0 > \mu > -1$,

and the loss rate seen by the process is constant, of unit rate, independent of the process. The closed form solution for the standard TCP controlled window evolution with constant loss rate is known from [2] as this corresponds to the case of AIMD protocol with constant loss rate.

5.2. Linear loss rate

We now consider the case when the loss rate seen by the window process depends linearly on the window process, i.e., $\lambda(w) = \lambda w$. The Kolmogorov equation for this system are

$$\pi(w) \frac{2\nu B w^{2+\mu}}{2 - B} = \int_{u=w}^{\frac{w}{B}} \lambda u \pi(u) du.$$

Defining another probability measure $\tilde{\pi}(w)$ by the transformation

$$\tilde{\pi}(w) = \frac{w\pi(w)}{\lambda E[W]}$$

along the lines of Section 3, we see that the above Kolmogorov equation becomes

$$\frac{2\nu B w^{1+\mu}}{(2-B)} \tilde{\pi}(w) = \int_{u=w}^{\frac{w}{B}} \tilde{\pi}(u) du.$$

Which is of the form of Eq. (11) with $1 + \mu$ replacing $2 + \mu$. Thus the method used to solve Kolmogorov eqs. (11) can be employed here as well; in particular, the above system of equations can be transformed to the one satisfied by standard AIMD protocol and results from [2] can be invoked.

Remark. The analysis of this section assumes that the multiplicative decrease factor is constant, equal to B . Though this assumption restricts the range of parameter choices we can make in order to tune HSTCP, the analysis gives important insights into dynamics of HSTCP controlled window evolution, for example, the importance of parameter μ . An approximation similar to ours has also been used in [5] to study router buffer behaviour under HSTCP controlled data transfer.

6. Numerical results

We obtained time average density of the window process from *ns-2* [13] simulations for AIMD protocol with constant loss rate and MIMD protocol with linear loss rate. The multiplicative decrease factor $\beta = 0.5$ for both the protocols and the loss rate, λ_a , for AIMD protocol was set to either 0.005 or 0.008. The MIMD protocol had an increase profile of $f_m(x) = 1.01x$ as in Scalable TCP while the AIMD protocol had $f_a(x) = 1$. The loss rate for MIMD protocol was $\lambda(x) = \lambda_m x$ where λ_m was chosen so that the conditions of Theorem 1 were satisfied. This requirement is satisfied if $\lambda_m = 0.01\lambda_a$, i.e., $\lambda_m = 0.00005$ or 0.00008 . Figure 4 gives the function $\pi_m(x)$ for MIMD and $\frac{C f_a(x) \pi_a(x)}{f_m(x)}$ where C is $\frac{\lambda_m E_m[X]}{\lambda_a}$ with $E_m[X]$ being the expected window size for MIMD protocol obtained from simulation. The results are as predicted by Theorem 1, i.e., $\pi_m(x) = \frac{\lambda_m E_m[X]}{\lambda_a}$, $\forall x$. For the same experimental setup, we also obtained the distribution of window sizes just before losses. The results are plotted in Figure 5 which shows that, in agreement with Theorem 2, this distribution is same for the two systems. Now, we compute the numerical values from our analysis of Section 4.2 and compare it with simulation results of Figure 4 for MIMD with linear loss rate. Figure 6 gives the comparison between analysis and simulations. Since the density function is already plotted in Figure 4, here we plot the $(E[X^n])^{\frac{1}{n}}$ vs. n for $1 \leq n \leq 10$. The analysis and simulations are seen to match well for smaller values of n ($n \leq 6$); the small discrepancy for large values of n could be owing to finite simulation run-length.

Figure 7 gives complementary distribution function of the stationary window process for HSTCP assuming that the multiplicative decrease factor $b(w)$ is fixed to a constant value B . Recall the parameters A , B , μ and ν of Section 5. We fix $A = 0$, $B = 0.125$ and ν so that $\frac{2B\nu}{2-B} = 0.01$ so that the case of $\mu = -1$ corresponds to the Scalable TCP [10]. The plot shows results for values of the parameter $\mu = -0.9, -1.0, -1.2$. In

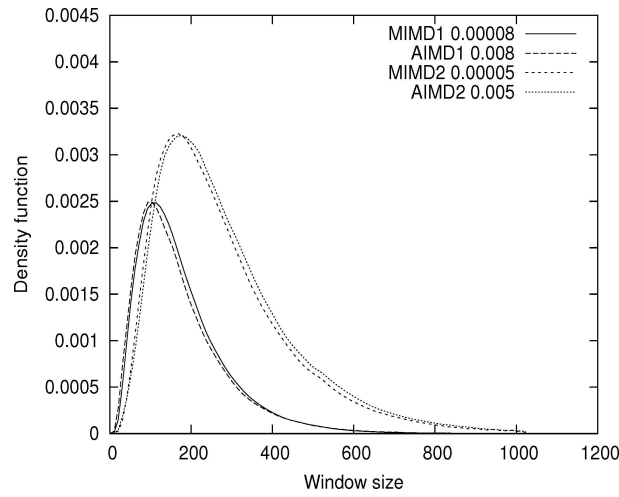


Figure 4. Comparison of time average distribution for MIMD with linear loss rate and for AIMD with constant loss rate.

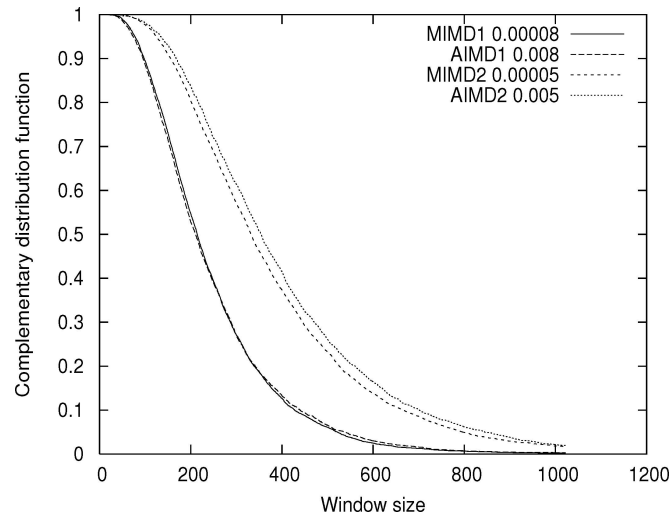


Figure 5. Window size distribution just before loss instants for MIMD with linear loss rate and for AIMD with constant loss rate.

order to do this, we varied the parameters p_l and p_h accordingly. The figure also gives numerical results from the analysis of Section 5. It is observed from the figure that one can approximate any increase function only by varying μ while keeping the multiplicative drop factor $b(w)$ constant. This simplifies the algorithm as now there are not many independent design choices and, moreover, the analysis of Section 5 combined with that of [2] provides closed form result for the stationary distribution. We also note that the distribution is very sensitive to the value of the parameter μ .

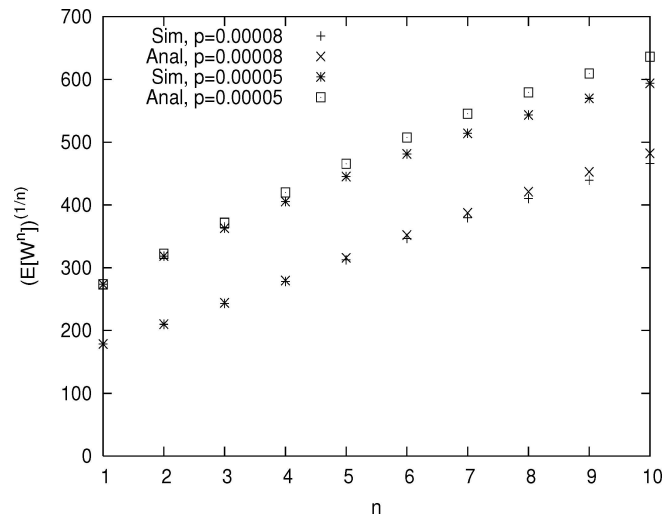


Figure 6. First 10 moments for MIMD with linear loss rate.

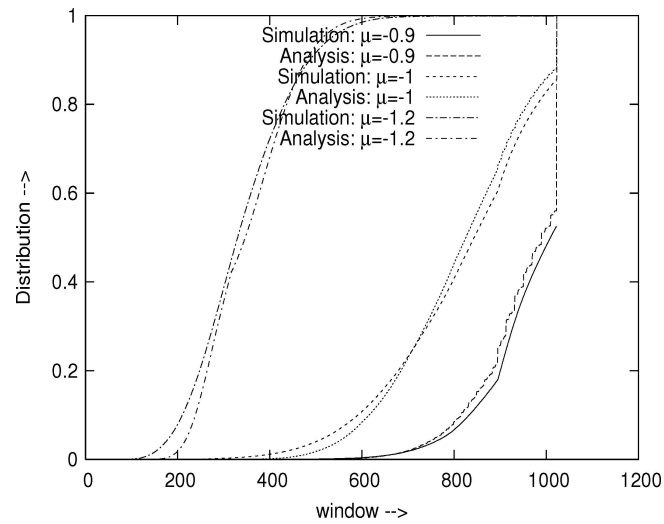


Figure 7. Window size distribution for HSTCP with linear loss rate.

7. Conclusion

We considered a general congestion control protocol with a state dependent loss probability. The Kolmogorov equations satisfied by the window evolution under this general setting were obtained. These equations were solved for specific TCP implementations of Scalable TCP and HighSpeed TCP under constant and linear loss rate assumptions. Various transformations introduced provided us with many equivalence relations. Most significant being that of the relation between window evolution and the workload process

in a finite capacity queueing system with state dependent service and arrival rates and a state dependent deterministic service requirement.

We have assumed that the loss rate, $\lambda(\cdot)$, is a given function. This may be the case in the applications using AQM schemes and where congestion losses are rare. This may also be the case when using very high speed links so that the packet losses in a round trip time are due to link layer losses. However, when most of the losses are owing to congestion losses, it appears to be more realistic that the form of $\lambda(\cdot)$ will itself be determined by the AWP. Also, it is possible that, like in model of [1], the loss process $\lambda(\cdot)$ may itself be a stochastic process. These considerations are topic of further research.

In the analysis of HSTCP we have chosen a multiplicative decrease algorithm with window independent decrease factor. We now aim at using some approximations for the evolution of the window process using the drop profile suggested in [8]. It is also important to study an optimal choice of the parameter μ that controls the behaviour of HSTCP.

Appendix

Consider an application using a general Adaptive Window Protocol. The evolution of the window size $\{x_t, t \geq 0\}$ at the end of the round trip times (RTT) can be described as follows:

$$x(t + RTT) = (x(t) + I(x(t)))I_{\{L(x(t))=0\}} + (x(t) + I_l(x(t)) - D(x(t) + r(x(t)))I_{\{L(x(t))=1\}}.$$

Here,

- RTT is the round trip time (assumed to be constant, corresponding to its average value).
- $L(x)$ is a $\{0, 1\}$ valued random variable which is 0 if there is no loss in an RTT starting with a window size of x and takes value of 1 when there is a loss in an RTT starting with window size of x . It is assumed that some form of the distribution of $L(\cdot)$ is given to us. An example of distribution of $L(\cdot)$ is that $P(L(x) = 1) = 1 - P(L(x) = 0) = \hat{p}x$ as used in [7].
- $I(x)$ is the amount by which the window size increases in event of no loss in an RTT starting with a window size of x . For the congestion avoidance phase of standard TCP, $I(x) = 1$. In this work we are assuming that $I(x) \ll x$. Note that this assumption need not be valid in general, for example in case of slow start phase of standard TCP, $I(x) = x$. However, in the protocols studied in this paper, this is indeed true.
- $D(x)$ is the amount by which the window is reduced in case of an event of a loss when window size is x . Note that $D(x)$ is not the window reduction in event of a loss in an RTT *starting* with a window of x . For the congestion avoidance phase of standard TCP, $D(x) = 0.5x$.

- $r(x)$ is such that $x + r(x) \leq x + I(x)$ is the window size at the instant of detection of loss event. In general, $r(x)$ will be a random variable over the interval $[0, I(x)]$.
- $I_l(x)$ is the *increase* in window size due to positive acknowledgements in an RTT. Note that $I_l(x)$ is a random variable over the interval $[0, I(x)]$. $I_l(x) = I(x)$ in the event of no loss.

To avoid consideration of randomness in $r(x)$ and $I_l(x)$, we have neglected $r(x)$ completely in the evolution (see [7] which also makes such an implicit assumption in the evolution equation). This can also be justified using our assumption that $I(x) \ll x$. We will also assume that $I_l(x) \approx I(x)$. This is true in the situation when not many packets are lost in an RTT. Hence we get the following approximation for the window evolution over RTTs,

$$x(t + RTT) = (x(t) + I(x(t)))(1 - I_{\{L(x(t))=1\}}) + (x(t) + I(x(t)) - D(x(t)))I_{\{L(x(t))=1\}}.$$

This evolution equation coincides with the approximation of [7] when we assume $I(x) = 1$ and $D(x) = 0.5x$. We then obtain,

$$x(t + RTT) - x(t) = I(x(t)) + D(x(t))I_{\{L(x(t))=1\}}.$$

Normalizing time by RTT, and observing that $I_{\{L(x(t))=1\}}$ is a $\{0, 1\}$ valued random variable, following the approach of [7], we approximate the distribution of $L(x(t))$ by that of $dN(t)$, where $\{N(t), t \geq 0\}$ is a Poisson process with a time varying stochastic rate function that depends on the value of the window size $x(t)$. We get the stochastic differential equation

$$dx(t) = I(x(t))dt + D(x(t))dN(t).$$

It is seen that the Kolmogorov equation can be written as, for $k \leq m - 2$:

$$\begin{aligned} \Pi'_k(y) &= \lambda(\Pi_{k+1}(y) - \Pi_k(y)) \\ \frac{d}{dy} \Pi_k(y) e^{\lambda y} &= \lambda e^{\lambda y} \Pi_{k+1}(y) \\ e^{\lambda y} \Pi_k(y) &= \lambda \int_{u_1=0+}^y e^{\lambda u_1} \Pi_{k+1}(u_1) du_1 + \Pi_k(0) \\ &= \lambda \int_{u_1=0+}^y e^{\lambda u_1} \Pi_{k+1}(u_1) du_1 + \Pi_k(0) \\ &= \lambda \int_{u_1=0+}^y \left[\lambda \int_{u_2=0+}^{u_1} e^{\lambda u_2} \Pi_{k+2}(u_2) du_2 + \Pi_{k+1}(0) \right] du_1 + \Pi_k(0) \\ &= \lambda \int_{u_1=0+}^y \left[\lambda \int_{u_2=0+}^{u_1} \left[\lambda \int_{u_3=0+}^{u_2} e^{\lambda u_3} \Pi_{k+3}(u_3) du_3 + \Pi_{k+2}(0) \right] du_2 \right] du_1 \end{aligned}$$

$$\begin{aligned}
& + \Pi_{k+1}(0) \Big] du_1 + \Pi_k(0) \\
= & \lambda^i \int_{u_1=0+}^y \int_{u_2=0+}^{u_1} \dots \int_{u_i=0+}^{u_{i-1}} e^{\lambda u_i} \Pi_{k+i}(u_i) du_i \dots du_2 du_1 \\
& + \sum_{j=0}^{i-1} \Pi_{k+j}(0) \frac{(\lambda y)^j}{j!} \quad \text{for } k+i \leq m-1.
\end{aligned}$$

In particular, for $i = m - k - 1$, we get

$$\begin{aligned}
e^{\lambda y} \Pi_k(y) = & \lambda^{m-k-1} \int_{u_1=0+}^y \int_{u_2=0+}^{u_1} \dots \int_{u_{m-k-1}=0+}^{u_{m-k-2}} e^{\lambda u_{m-k-1}} \Pi_{m-1}(u_{m-k-1}) du_{m-k-1} \dots du_2 du_1 \\
& + \sum_{j=0}^{m-k-2} \Pi_{k+j}(0) \frac{(\lambda y)^j}{j!}. \tag{12}
\end{aligned}$$

For $(m-1)\theta \leq y$, the up and down crossing rates are equated as follows:

$$\begin{aligned}
\Pi'(y) &= \lambda \int_{u=y+}^{\infty} \pi(u) du \\
\Pi'(y) &= \lambda(1 - \Pi(y)) \\
\Pi'_{m-1}(y) &= \lambda(1 - \Pi_{m-1}(y)) \\
\frac{d}{dy} \Pi_{m-1}(y) e^{\lambda y} &= \lambda e^{\lambda y} \\
\Pi_{m-1}(y) e^{\lambda y} &= \int_{u=0}^y \lambda e^{\lambda u} du + \Pi_{m-1}(0) \\
&= e^{\lambda y} - 1 + \Pi_{m-1}(0).
\end{aligned}$$

Substituting this in Eq. (12), we get,

$$\begin{aligned}
e^{\lambda y} \Pi_k(y) = & \lambda^{m-k-1} \int_{u_1=0+}^y \int_{u_2=0+}^{u_1} \dots \int_{u_{m-k-1}=0+}^{u_{m-k-2}} e^{\lambda u_{m-k-1}} du_{m-k-1} \dots du_2 du_1 \\
& + \sum_{j=0}^{m-k-1} \Pi_{k+j}(0) \frac{(\lambda y)^j}{j!} - \frac{(\lambda y)^{m-k-1}}{(m-k-1)!} \\
= & e^{\lambda y} - \sum_{j=0}^{m-k-1} (1 - \Pi_{k+j}(0)) \frac{(\lambda y)^j}{j!}. \tag{13}
\end{aligned}$$

Now, noting that $\Pi_k(0) = \Pi_{k-1}(\theta)$ and that $\Pi(0) = \Pi_0(0) = 0$, and integrating the above, we get a value of $\Pi((m-1)\theta)$. $\Pi(0) = \Pi_0(0) = 0$ because in any visit to this point, the window instantaneously attains a positive value with probability 1 because of

constant window increase rate and since otherwise we would require uncountably many Poisson instants occurring in a continuum of time.

The original system of MIMD can be obtained by the reverse transformation. Proposition 1 thus follows.

We know from Eq. (13) that

$$e^{\lambda y} \Pi_k^c(y) = \sum_{j=0}^{m-k-1} \Pi_{k+j}^c(0) \frac{(\lambda y)^j}{j!}, \quad k \leq m-2.$$

Let $F_k \triangleq \Pi_k^c(0)$. From continuity of $\Pi(\cdot)$, it follows that

$$F_{k+1} = 1 - \Pi_{k+1}(0) = 1 - \Pi_k(\theta).$$

Thus,

$$\begin{aligned} e^{\lambda \theta} F_{k+1} &= \sum_{j=0}^{m-k-1} F_{k+j} \frac{(\lambda \theta)^j}{j!} \\ \Rightarrow F_k &= a F_{k+1} - \sum_{j=1}^{m-k-1} F_{k+j} \frac{b^j}{j!}, \quad k \leq m-2, \end{aligned}$$

where $b = \lambda \theta$ and $a = e^b$. The above relation can be applied again to get

$$\begin{aligned} F_k &= a F_{k+1} - \sum_{j=1}^{m-k-1} F_{k+j} \frac{b^j}{j!} \\ &= a^2 F_{k+2} - a \sum_{j=1}^{m-k-2} F_{k+1+j} \frac{b^j}{j!} - \sum_{j=1}^{m-k-1} F_{k+j} \frac{b^j}{j!} \\ &= a^l F_{k+l} - \sum_{s=0}^{l-1} a^s \sum_{j=1}^{m-k-1-s} F_{k+s+j} \frac{b^j}{j!}, \quad l \leq m-k-1, \quad (k+l-1 \leq m-2) \\ &= a^{m-k-1} F_{m-1} - \sum_{s=0}^{m-k-2} a^s \sum_{j=1}^{m-k-1-s} F_{k+s+j} \frac{b^j}{j!} \\ &= a^{m-k-1} F_{m-1} - \sum_{l=1}^{m-k-1} F_{k+1} \sum_{j=1}^l a^l \frac{(\frac{b}{a})^j}{j!} \\ &= a^{m-k-1} F_{m-1} - \sum_{l=1}^{m-k-1} F_{k+1} \phi_1(l) \\ &= (a^{m-k-1} - \phi_1(m-k-1)) F_{m-1} - \sum_{l=1}^{m-k-2} F_{k+l} \phi_1(l), \end{aligned} \tag{14}$$

where $\phi_1(l) \triangleq \sum_{j=1}^l a^l \frac{b^j}{j!}$ is independent of k . Note that $\phi(1) = b$, implying $F_{m-2} = F_{m-1}(a - b)$. Using Equation (14) again, we get, for $k \leq m - 3$,

$$\begin{aligned}
F_k &= (a^{m-k-1} - \phi_1(m-k-1))F_{m-1} \\
&\quad - \sum_{l=1}^{m-k-2} \phi_1(l)[(a^{m-k-l-1} - \phi_1(m-k-l-1))F_{m-1} - \sum_{s=1}^{m-k-l-2} F_{k+l+s}\phi_1(s)] \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] \\
&\quad + \sum_{l=1}^{m-k-2} \phi_1(l) \sum_{s=1}^{m-k-l-2} F_{k+l+s}\phi_1(s) \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] \\
&\quad + \sum_{l=1}^{m-k-3} \phi_1(l) \sum_{s=1}^{m-k-l-2} F_{k+l+s}\phi_1(s) \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] \\
&\quad + \sum_{l=2}^{m-k-2} F_{k+l} \sum_{s=1}^{l-1} \phi_1(l-s)\phi_1(s) \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
&\quad - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] + \sum_{l=2}^{m-k-2} F_{k+l}\phi_2(l) \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
&\quad - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] \\
&\quad + \sum_{l=2}^{m-k-2} \phi_2(l)[(a^{m-k-l-1} - \phi_1(m-k-l-1))F_{m-1} - \sum_{s=1}^{m-k-l-2} F_{k+l+s}\phi_1(s)] \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
&\quad - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] + \sum_{l=2}^{m-k-2} \phi_2(l)(a^{m-k-l-1} \\
&\quad - \phi_1(m-k-l-1))] - \sum_{l=2}^{k-k-2} \phi_2(l) \sum_{s=1}^{m-k-l-2} F_{k+l+s}\phi_1(s) \\
&= F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
&\quad - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))]
\end{aligned}$$

$$\begin{aligned}
& + \sum_{l=2}^{m-k-2} \phi_2(l)(a^{m-k-l-1} - \phi_1(m-k-l-1)) \\
& - \sum_{l=2}^{m-k-3} \phi_2(l) \sum_{s=1}^{m-k-l-2} F_{k+l+s} \phi_1(s) \\
= & F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
& - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1)) \\
& + \sum_{l=2}^{m-k-2} \phi_2(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] \\
& - \sum_{l=3}^{m-k-2} F_{k+l} \sum_{s=1}^{l-2} \phi_2(l-s)\phi_1(s) \\
= & F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
& - \sum_{l=1}^{m-k-2} \phi_1(l)(a^{m-k-l-1} - \phi_1(m-k-l-1)) \\
& + \sum_{l=2}^{m-k-2} \phi_2(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] - \sum_{l=3}^{m-k-2} F_{k+l}\phi_3(l) \\
= & F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
& + \sum_{s=1}^j (-1)^s \sum_{l=s}^{m-k-2} \phi_s(l)(a^{m-k-l-1} - \phi_1(m-k-l-1))] \\
& + (-1)^{j+1} \sum_{l=j+1}^{m-k-2} F_{k+l}\phi_{j+1}(l), \quad j+1 \leq m-k-2 \\
= & F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) \\
& + \sum_{s=1}^{m-k-3} (-1)^s \sum_{l=s}^{m-k-2} \phi_s(l)(a^{m-k-l-1} - \phi_1(m-k-l-1)) \\
& + (-1)^{m-k-2} F_{m-2}\phi_{m-k-2}(m-k-2) \\
\Rightarrow F_k = & F_{m-1}[(a^{m-k-1} - \phi_1(m-k-1)) + \sum_{s=1}^{m-k-3} (-1)^s \\
& \times \sum_{l=s}^{m-k-2} \phi_s(l)(a^{m-k-l-1} - \phi_1(m-k-l-1)) \\
& + , (-1)^{m-k-2}(a-b)\phi_{m-k-2}(m-k-2)].
\end{aligned}$$

Here,

$$\phi_{j+1}(l) = \sum_{s=1}^{l-j} \phi_j(l-s)\phi_1(s), \quad j \geq 1.$$

The above expression for F_k valid for $k \leq m - 2$ if we define $\phi_0(0) = 0$. Now, since $F_0 = 1$, we see that

$$\begin{aligned}
 1 &= F_{m-1}[(a^{m-1} - \phi_1(m-1)) - \sum_{s=1}^{m-3} (-1)^s \sum_{l=s}^{m-2} \phi_s(l)(a^{m-l-1} - \phi_1(m-l-1)) \\
 &\quad + (-1)^{m-2}(a-b)\phi_{m-2}(m-2)] \\
 F_{m-1} &= [(a^{m-1} - \phi_1(m-1)) - \sum_{s=1}^{m-3} (-1)^s \sum_{l=s}^{m-2} \phi_s(l)(a^{m-l-1} - \phi_1(m-l-1)) \\
 &\quad + (-1)^{m-2}(a-b)\phi_{m-2}(m-2)]^{-1}.
 \end{aligned}$$

This proves Proposition 2.

For $x \in I_{m-k}, k \geq 2$, we get the following Kolmogorov equation

$$\pi(x)x = \begin{cases} \int_{\frac{x}{\beta}}^x \pi(u)\lambda u du & x \in I_{m-k}, k \geq 2, \\ \int_{u=x}^{u=\frac{x}{\beta}} \pi(u)\lambda u du + \tilde{P}_M \lambda M & x \in I_{m-1}. \end{cases}$$

Let $E[X] = \int_{x=1}^M \pi(x)x dx - P_M M$. Dividing both sides of the above Kolmogorov equation by $E[X]$ and defining $\tilde{\pi}(x) = \frac{x\pi(x)}{E[X]}$ and $\tilde{P}_M = \frac{P_M M}{E[X]}$, we get

$$\tilde{\pi}(x) = \begin{cases} \int_{\frac{x}{\beta}}^x \tilde{\pi}(u) \lambda du & x \in I_{m-k}, k \geq 2, \\ \int_{u=x}^{u=\frac{x}{\beta}} \tilde{\pi}(u)\lambda du + P_M \lambda & x \in I_{m-1}. \end{cases}$$

This is the Kolmogorov equation for AIMD protocol under constant loss rate analysed in [2]. The difference is that here the slope of linear increase is unity instead of the parameter α in [2]. We know from [2] that the solution to above Kolmogorov equations is (the complementary distribution function)

$$\tilde{\pi}(x) = \tilde{P}_M \sum_{i=1}^k \frac{c_i^{(k)} \lambda}{\beta^{i-1}} e^{-x \frac{\lambda}{\beta^{i-1}}},$$

where

$$c_{i+1}^{(k)} = \frac{c_i^{(k-1)}}{1 - \beta^{-i}}$$

and

$$c_1^{(k)} = e^{\lambda M \beta^{k-1}} \left[\sum_{i=1}^{k-1} e_i^{(k-1)} e^{-\lambda M \beta^{k-i}} - \sum_{i=2}^k c_i^{(k)} e^{-\lambda M \beta^{k-i}} \right]$$

and

$$\tilde{P}_M = \left[\sum_{i=1}^m e_i^{(k)} e^{-\lambda M \beta^{k-i}} \right]^{-1}.$$

Hence, for $x \in I_{m-k}$,

$$\pi(x) = \tilde{P}_M E[X] \sum_{i=1}^k \frac{c_i^{(k)} \lambda}{\beta^{i-1}} \frac{e^{-x \frac{\lambda}{\beta^{i-1}}}}{x},$$

and

$$\begin{aligned} \Pi^c(x) &= \int_{u=x}^{M^-} \pi(u) du - P_M \\ &= \int_{u=x}^{M\beta^{k-1}} \pi(u) du + \Pi^c(M\beta^{k-1}) \\ &= \tilde{P}_M E[X] \sum_{i=1}^k \frac{c_i^{(k)} \lambda}{\beta^{i-1}} \int_{u=x}^{M\beta^{k-1}} \frac{e^{-x \frac{\lambda}{\beta^{i-1}}}}{x} dx + \Pi^c(M\beta^{k-1}) \\ &= \tilde{P}_M E[X] \sum_{i=1}^k \frac{c_i^{(k)} \lambda}{\beta^{i-1}} \Gamma\left(0, \frac{x\lambda}{\beta^{i-1}}, \frac{M\lambda}{\beta^{i-k}}\right) + \Pi^c(M\beta^{k-1}) \\ \Pi^c(M\beta^k) &= \tilde{P}_M E[X] \sum_{i=1}^k \frac{c_i^{(k)} \lambda}{\beta^{i-1}} \Gamma\left(\frac{M\lambda}{\beta^{i-1-k}}, \frac{M\lambda}{\beta^{i-k}}\right) + \Pi^c(M\beta^{k-1}). \end{aligned}$$

Where $\Gamma(0, a, b) = \int_{t=a}^b \frac{e^{-t}}{t} dt$ is the difference of the upper incomplete Gamma functions, $\Gamma(0, a, b) = \Gamma(0, a) - \Gamma(0, b)$ where $\Gamma(0, a) = \int_{t=a}^{\infty} \frac{e^{-t}}{t} dt$. Now,

$$\begin{aligned} \Pi^c(M\beta) &= \tilde{P}_M E[X] c_1^{(1)} \lambda \Gamma(0, M\lambda\beta, M\lambda) + P_M \\ &= \tilde{P}_M E[X] c_1^{(1)} \lambda \Gamma(0, M\lambda\beta, M\lambda) + \tilde{P}_M \frac{E[X]}{M}. \end{aligned}$$

Thus we find $\Pi^c(M\beta^k)$, $k \geq 1$ in terms of $E[X]$ since we know the other quantities in the above expressions. Now, since $\Pi^c(1) = \Pi^c(M\beta^m) = 1$, we get the value of $E[X]$, hence $\pi(\cdot)$ for all values of x . Proposition 3 thus follows.

Acknowledgments

This work was supported by grant from the *Centre Franco-Indien pour la Promotion de la Recherche Avancee* (CEFIPRA) under project no. 2900-IT-1. This work was also partially supported by the European Network of Excellence EURO NGI.

References

- [1] E. Altman, K. Avrachenkov and C. Barakat, A stochastic model of TCP/IP with stationary random losses, in: *Proc. of the ACM SIGCOMM*. (2000).
- [2] E. Altman, K. Avratchenkov, C. Barakat and R. Núñez Queija, State-dependent M/G/1 type queueing analysis for congestion control in data networks, in: *Proc. of IEEE INFOCOM*. (2001).
- [3] S. Asmussen, *Applied Probability and Queues*, (Springer 2003).
- [4] F. Baccelli and D. Hong, AIMD, fairness and fractal scaling of TCP traffic, in: *Proc. of the IEEE INFOCOM* (2002).
- [5] D. Barman, G. Smaragdakis and I. Matta, The effect of router buffer size on highspeed TCP, in: *Proceedings of Globecom 2004*, (2004).
- [6] R. Bekker, S.C. Borst, O.J. Boxma and O. Kella, Queues with workload-dependent arrival and service rates, *Queueing Sys.* 46(34) (2004) 537–556.
- [7] A. Budhiraja, F. Hernández-Campos, V.G. Kulkarni and F. D. Smith, Stochastic differential equation for TCP window size: Analysis and experimental validation. *Prob. in the Engg. and Informational Sciences* 18 (2004) 111–140.
- [8] S. Floyd, *Highspeed TCP for Large Congestion Windows*. (RFC 3649. Experimental, 2003).
- [9] V. Jacobson, Congestion avoidance and control. *ACM SIGCOMM* 88 (1988).
- [10] T. Kelly, Scalable TCP: Improving performance in highspeed wide area networks, *Computer Comm. Review.* 33(2) (2003) 83–91.
- [11] A.A. Kherani and A. Kumar, Stochastic models for throughput analysis of randomly arriving elastic flows in the internet, in: *Proc. of IEEE INFOCOM*. (2002).
- [12] L. Kleinrock, *Queueing Systems Volume 1: Theory* (Wiley & sons. 1975).
- [13] S. McCanne and S. Floyd, *ns: Network Simulator*. Available at <http://www.isi.edu/nsnam/ns/>.
- [14] T.J. Ott, J.H.B. Kemperman and M. Matins, *The Stationary Behavior of Ideal TCP Congestion Avoidance*. Unpublished manuscript (1996).
- [15] D. Perry, W. Stadje and S. Zacks, The M/G/1 queue with finite workload capacity, *Queueing Sys.* 39(1) (2001) 7–22.