Markov Processes with State-Dependent Failure Rates and Application to RED and TCP Window Dynamics

Archan Misra IBM
IBM T.J. Watson Res. Center
19 Skyline Drive
Hawthorne, NY 10532, USA

archan@us.ibm.com

Teunis J. Ott
Computer Science Dept.
New Jersey Institute
of Technology
Newark, NJ, USA

ott@oak.njit.edu

John S. Baras*
Electrical and Computer
Engineering Dept. and the
Institute for Systems Research
University of Maryland
College Park, MD 20742, USA
baras@isr.umd.edu

***Abstract*— This paper presents a mathematical technique for computing the stationary distribution of Markov processes that evolve deterministically between arbitrarily distributed 'failure events'. The key innovation in this paper is the use of a state-dependent time re-scaling technique, such that the re-scaled process can be described by a Poisson-interrupted stochastic differential equation. This technique is first applied to compute the stationary window distribution of a TCP flow performing idealized classical congestion avoidance under variable, but state-dependent, packet loss, and subsequently, to study the distribution of a TCP flow performing generalized congestion avoidance. We show how the stochastic differential equation can be solved by a rapidly convergent numerical technique to obtain the stationary distribution in the re-scaled (subjective) time, and present the re-scalings needed to eventually obtain the distribution of the original Markov process. We demonstrate how this analysis can be used to compute the window distribution of a TCP flow interacting with a RED, ERD or ECN queue, with or without minimally assured throughput guarantees.**

I. Introduction

In this paper, we analyze the stationary distribution of a class of feedback-controlled Markov processes, where the feedback events occur with a *random* but *state-dependent* probability. In other words, the state transitions of the process occur with a state-dependent probability, but are *conditionally* independent of past and future transition events. This research was motivated by a desire to study the stationary distribution of TCP (the Transmission Control Protocol), which is, by far, the most dominant adaptive transport protocol used to regulate Internet traffic. In the stationary phase, TCP regulates the injection of new packets using a 'congestion avoidance' algorithm [1], whereby the congestion window ($cwnd$) is increased only on successful reception of an 'acknowledgment' packet (positive feedback) and decreased on determination of a missing acknowledgment (negative feedback). Internet routers provide this feedback through either randomized packet drops or randomized packet marking techniques, with the feedback rate (dropping/marking probability) a function, *directly, of the router queue occupancy, and, indirectly, of the congestion window size.* Our analytical contributions can thus be viewed as an extension to earlier work on TCP analysis (e.g., [2]), where the TCP congestion window is computed under the assumption of a *constant* feedback rate.

We consider one-dimensional Markovian processes that evolve deterministically between the occurrence of 'failure events', with each inter-failure duration dependent only on the process state since the last failure event (and independent of all past and future failure events). *We do not impose any specific distribution on the inter-failure duration.* This stochastic model applies to TCP behavior when it is abstracted into a continuous cycle of 'congestion avoidance', packet loss/marking and 'fast recovery' [3]. We disregard the details of TCP timeouts and fast recovery and assume an idealized behavior, whereby a congestion notification that occurs when the congestion window is W Maximum Segment Sizes (MSSs) *instantaneously* reduces the congestion window (and the number of unacknowledged packets) to $\lceil \frac{W}{2} \rceil$ MSSs. The dynamics of TCP window evolution can then be captured by a discrete-time Markov process with state-dependent transition probabilities. To further demonstrate the utility of our mathematical technique, we also consider a more general class of TCP-like *generalized* congestion avoidance algorithms. This is a parametric generalization of conventional TCP congestion avoidance and belongs to the class of binomial congestion control algorithms that have been studied recently [4]. More importantly, recent research (e.g., [5]) has demonstrated how such parametrized modifications to TCP behavior can lead to higher network utilization and lower variation in queue sizes in the emerging QoS-aware and ECN-capable Internet. Under generalized congestion avoidance, TCP increases the $cwnd$ from its current value W by $c_1 W^{\alpha}$ on receiving an acknowledgment without congestion indication (packet drop or marking) and decreases it by $c_2 W^{\beta}$ on receiving an acknowledgment containing a congestion indicator. Here α, β, c_1 and c_2 are constants that parametrize the algorithm; clearly, choosing $\alpha = -1$, $\beta = 1$, $c_1 = 1$ and $c_2 = 0.5$ results in the classical (TCP) congestion avoidance algorithm.

*Research supported in part by DARPA contract No. N66001-00-C-8063

While the TCP window evolution belongs to the class of discrete-space (countably infinite), discrete-time processes, our analytical technique applies to the more general case of a continuous-time, continuous-space process $W(t)$. The key innovation in our analysis is the employment of state-dependent rescaling in both the time and space axes. In particular, we study the properties of a new process $Y(\tau)$, derived from $W(t)$, where the time index τ is a non-linear function of t. By using an appropriate time rescaling function, the 'points of failure' of the process $Y(\tau)$ become realizations of a Poisson process, thereby allowing us to relate the steady-state probabilities via a Kolmogorov equation. The differential equation is then solved through a novel iterative numerical technique, which can be shown to exhibit rapid and guaranteed numerical convergence.

For ease of exposition, we shall usually restrict our formulation and notations to the case of TCP congestion avoidance (ideal or generalized), taking care to point out the modifications needed for more generic stochastic processes. Accordingly, we consider the stochastic process $(W_n)_{n=1}^{\infty}$, where W_n stands for the congestion window just after the n^{th} acknowledgment packet has arrived at the source. The resulting discrete-time Markov process exhibits the following conditional probabilities:

$$P\{W_{n+1} = w + c_1 w^{\alpha} | W_n = w\} = 1 - p(w) \qquad (1)$$

$$P\{W_{n+1} = w - c_2 w^{\beta} | W_n = w\} = p(w), \qquad (2)$$

where $p(w)$ is the congestion notification probability when the congestion window is w. (The time index n in the above equations is referred to as *ack time* in this paper, since it increases only with the receipt of acknowledgments.) In the TCP case, the 'points of failure' of the processes $W(t)$ and $Y(\tau)$ correspond to the receipt of congestion feedback from the routers in the traffic path. Such feedback is usually provided through either randomized packet dropping (e.g., the Randomized Early Detection (RED) [6]) or through explicit packet marking (e.g., the Explicit Congestion Notification (ECN) mechanism [7]). The exact feedback mechanism is unimportant for our analysis, which considers TCP response to abstract congestion notifications and does not distinguish between packet dropping and marking mechanisms. We shall, however, provide simulation results with both dropping and marking based service models to evaluate the accuracy of our analytical technique.

The rest of the paper is organized as follows. In section II, we provide a survey of related work and also discuss the applicability of our model to TCP traffic. In section III, we describe the time and space rescalings, as applied to both TCP performing *classical* congestion avoidance, and to a more generic class of Markov processes. In section IV, we obtain the resulting Kolmogorov equation for this re-scaled process, and derive the iterative technique for rapidly solving this differential equation. Section V provides numerical examples analyzing the window behavior of TCP classical congestion avoidance with Early Random Drop and Random Early Detection queues and evaluates the effectiveness of our numerical techniques in predicting TCP behavior. While section VI shows how the numerical technique applies to the more general case of a TCP process performing *generalized* (as opposed to classical) congestion avoidance, section VII applies this analysis to the interaction of a generalized TCP flow with an ORED [8] buffer under the Assured Service [9] model. Finally, section VIII concludes the paper.

II. Related Work and Model Applicability

There has been a fairly large body of literature analyzing the dynamics of TCP congestion control. All of the early papers, however, assume a constant drop or marking probability. The 'square-root' formula, which states that the average window of a persistent TCP connection if of the order $\sqrt{p}$, and which ignores the effects of TCP timeouts and fast recovery, has been rigorously derived in [2] and, less rigorously, in [10] and [11] (the last publication also considers modifications to the formula resulting from losses of acknowledgment packets). By considering the effects of fast recovery and timeouts in greater detail for various TCP versions, [12], [13] provide better estimates of throughput (especially at larger loss probabilities). Among these papers, only [2] derives the *stationary window distribution* of the TCP flow, albeit for a constant notification probability p. [2] employs a scaling technique, where the time axis is rescaled linearly by a factor p, and the state space is rescaled linearly by a factor $\sqrt{p}$, resulting in a rescaled process $W(t) = \sqrt{p} W_{\lfloor \frac{t}{p} \rfloor}$. (We call the time index generated by the rescaling *subjective time*.) We shall also employ similar rescalings in this paper. While our space rescaling will still be linear, the variable loss probability of our model requires the time rescaling to be non-linear, as explained in Section III.

To evaluate the accuracy of our mathematical technique, we shall compare the analytical model against simulation studies performed with popular TCP versions (Reno and NewReno). The individual TCP flow is subject to packet drops performed by a router buffer according to the popular Random Early Detection (RED) or *Early Random Drop* (ERD) [19] algorithms. In an ERD buffer, the drop probability is a function of the *instantaneous* buffer occupancy; in a RED buffer, the drop probability is a function of the *average* queue length.

To further study the applicability of our analysis to generalized TCP congestion avoidance, we shall also analyze the interaction of a generalized TCP flow with a router buffer under the more complicated Assured Service [9] model. Under this model, a TCP flow is associated with a minimum *assured rate* and is subject to congestion notification only when it exceeds this rate. We consider the interaction with an ORED buffer, which is described in [8], and which essentially randomly marks packets (similar to ECN), but only if they have been *tagged* as non-conformant at the network edge. The reasonable accuracy of our analytical model demonstrates the practical utility of our mathematical technique.

III. Process Model and Rescalings

In this section, we first describe the discrete-time model for TCP classical congestion avoidance and provide the appropriate time and space rescalings used to derive a more amenable continuous-time, continuous-space process characterized by Poisson points of failure.

A. The Model for TCP Behavior

The TCP source is assumed to send a large data file in the forward direction with the congestion window acting as the only constraint on the transmission of packets. It is assumed that the connection never goes into timeout, that the receive or advertised window never limits the number of unacknowledged packets, that data is always sent in equal-sized segments (one MSS) and that acknowledgments are never lost. The receiver generates an acknowledgment for every received packet (we shall also extend the analysis to model the phenomenon of 'delayed acknowledgments'). Packet losses are assumed to be conditionally independent.

For the case of classical TCP congestion avoidance, equations (1) and (2) reduce to:

$$P\{W_{n+1} = w + \frac{1}{w} | W_n = w\} = 1 - p(w), \tag{3}$$

$$P\{W_{n+1} = \frac{w}{2} | W_n = w\} = p(w), \tag{4}$$

where p is the packet dropping probability.

The time index in equations (3) & (4) is called *ack time* and is is a positive-integer valued variable that increments by 1 whenever an acknowledgment packet arrives at the source. Ack time increases linearly with clock time only when the window size and round trip times are both constant. Let the cumulative probability stationary distribution for this process under this ack time be $F_{ack}(.)$.

B. Time and State-space Rescaling

To derive a more amenable continuous-time, continuous-valued random process from the process described by equations (3) & (4), we rescale both the time and state-space axes. This leads us to introduce the concept of *subjective time*, which is, roughly speaking, related to ack time through an invertible mapping. For the case considered in [2], where the loss probability was a constant p, the subjective time was derived from ack time by *linearly compressing* the time scale by a factor p, by using the relation $dt_{subjective} = p.dt_{ack}$. When the loss probability is not constant but state-dependent, a state-dependent (non-linear) scaling must be used.

For the specific TCP process under consideration, our quantized increment in subjective time t is provided by the mapping

$$\Delta t \ = \ p(W_n)\Delta n \tag{5}$$

where Δt is the (real-valued) increment in subjective time, Δn is the (integer-valued) increment in ack time and $p(W_n)$ is the loss probability associated with the value of the window W_n at ack time n. *In other words, for a process defined under this subjective time, time advances at a variable rate, as an increase in the ack time index of* 1 *corresponds to a state-dependent increase of* $p(W_n)$ *in the subjective time index.* Thus, $t(N)$, the subjective time immediately after sending packet number N, is expressed as $t(N) = \sum_{i=1}^{N} p(W_i)$. As $0 \leq p(W_n) \leq 1$, t is a real-valued sequence obtained by a contraction of the ack time index. As $p_{max} \downarrow 0$, the limiting subjective time index becomes a continuous variable. We shall see that, for this specific case, *the process defined in subjective time has a failure rate that becomes Poisson and constant asymptotically*, as the maximum dropping probability $p_{max} \downarrow 0$.

If $W'(t)$ represents the process W_n in subjective time t via the transformation in equation (5), its sample path between the events of packet failure can be modeled by the difference equation

$$\frac{\Delta W'}{\Delta t} \ = \frac{1}{p(W')W'} \tag{6}$$

As $p_{max} \downarrow 0$, the difference equation can be modeled by a corresponding differential equation with increasing accuracy. The differential equation would however, in the limit, be ill-behaved as the derivative goes to ∞ as $p_{max} \downarrow 0$. To obtain a well behaved process, we also need to rescale the state space of $W'(t)$. To rescale properly, we assume that $\frac{p(W)}{p_{max}} > \epsilon \ \forall W$, (i.e., the ratio between the minimum and maximum loss probabilities is uniformly bounded away from 0). If we then rescale the state-space of the process $W'(t)$ by the multiplicative constant $\sqrt{p_{max}}$, the resulting process, which we call $W(t)$, obeys the functional relationship

$$W(t) = \sqrt{p_{max}} \ W_n, \tag{7}$$

$$where \ n = \ n(t) \ = \ arg \ max \ j : \ \sum_{i=0}^{j} p(W_i) \leq \ t$$

This continuous-time and continuous valued process $W(t)$ will be the subject of our study and analysis.

Theorem 1: *As $p_{max} \downarrow 0$, the process defined by equations (3), (4) & (7), converges (path-wise) to a process whose window, $W(t)$, behaves as follows:*
There is a Poisson process with intensity 1*, the points of which are denoted by $(\tau_n)_{n=1}^{\infty}$. In between the points of this Poisson process, the window, W, evolves according to the equation*

$$\frac{dW}{dt} = \frac{p_{max}}{p(\frac{W}{\sqrt{p_{max}}})W} = \frac{1}{q(W)} \tag{8}$$

At the points of the realization of the Poisson process, we have $W(\tau^+) = \frac{1}{2}W(\tau^-)$.

C. Distribution in (Continuous) ACK Time

We shall see how to compute $F_{subj}(w)$, the stationary cumulative distribution of $W(t)$ in subjective time, later in section IV. We now consider how to correct this distribution for the state-space and time rescalings, introduced in equation (7), assuming $F_{subj}(w)$ is already known.

The state-space scaling results in a simple linear transformation of the probability distribution. $F_{subj}(w)$ is corrected first to obtain $F_s(w)$, the cumulative stationary distribution in subjective time but without space-rescaling by the relationship $F_s(w) = F_{subj}(\sqrt{p_{max}}w)$.

The sampling non-uniformity due to time-scaling is corrected, to obtain $F_{ack}(w)$, by dividing the probability density in subjective time, $dF_s(w)$, by the appropriate quantity $p(w)$. This is achieved by the transformation

$$dF_{ack}(w) = \frac{\frac{dF_s(w)}{p(w)}}{\int \frac{dF_s(\eta)}{p(\eta)}} \tag{9}$$

IV. The Stationary Kolmogorov Equation and its Solution

In this section we obtain the stationary distribution of the process, defined in section III.B, whose behavior is described by the equation $\frac{dW(t)}{dt} = \frac{1}{q(W(t))}$ in between the points of a Poisson process of rate λ. At the points of the Poisson process, $W(t)$ is obtained by $W(t^+) = A(W(t^-))$; let $a(x)$ be the inverse function of $A(x)$.

Theorem 2: *The stationary cumulative distribution $F_{subj}(x)$ of the process in section III.B satisfies the differential equation*

$$\frac{dF_{subj}(x)}{dx} = \lambda q(x)(F_{subj}(a(x)) - F_{subj}(x)) \tag{10}$$

We were unable to obtain a closed-form analytical solution for this differential equation. We however provide an open-form analytical expression for $F_{subj}(x)$ that translates into a rapidly converging numerical technique for evaluating the cumulative distribution. In passing, we note that the approximation of the TCP process results in the differential equation

$$\frac{dF_{subj}(x)}{dx} = q(x)(F_{subj}(2x) - F_{subj}(x)), \tag{11}$$

which will be used in the numerical examples to be presented later.

A. Solution of the Equation

Let G be the complementary distribution function defined by the relation $G(x) = 1 - F_{subj}(x)$. Equation (10) is equivalent to the equation

$$\frac{dG(x)}{dx} + \lambda q(x)G(x) = \lambda q(x)G(a(x)) \tag{12}$$

with the boundary conditions $G(0) = 1$, $G(\infty) = 0$. Let $Q(x) = \int_0^x \lambda q(u)du$ and define $G(x) = H(x)e^{-Q(x)}$ where $H(x)$ is an arbitrary function (to be evaluated). $H(x)$ is then seen to obey the differential equation

$$H(x) = H(z) - \lambda \int_x^z q(u)e^{Q(u)}G(a(u))du \tag{13}$$

By letting $z \uparrow \infty$ in equation (13) and noting that $G(a(u)) = e^{-Q(a(u))}H(a(u))$, we have

$$H(x) = \bar{H} - \lambda \int_x^\infty q(u)e^{(Q(u)-Q(a(u)))}H(a(u))du \tag{14}$$

with the boundary conditions $H(0) = 1$ and $H(\infty) = \bar{H}$.

By defining $J(u)$ as $J(u) = \lambda q(u)e^{Q(u)-Q(a(u))} = \lambda q(u)e^{-\int_u^{a(u)} q(\rho)d\rho}$, equation (14) reduces to

$$H(x) = \bar{H} - \int_x^\infty J(u)H(a(u))du \tag{15}$$

B. Numerical Computation

Repeated substitution in equation (15) offers a numerical technique for evaluating $H(x)$. As $H(x)$ tends to a limit as $x \uparrow \infty$, it can be treated as a constant beyond a certain value x_{upper} (chosen such that the resulting error in computing $H(x)$ is at most a small value ϵ). We can then obtain an approximation for $H(x)$ by setting the value of $H(x)$ beyond x_{upper} to be a constant and computing $H(x)$ between $(0, x_{upper})$. After the algorithm converges, we can divide by $H(0)$ to satisfy the boundary conditions $H(0) = 1, H(\infty) = \bar{H}$.

The complete numerical procedure for computing $F_{subj}(x)$ is as follows:

1) Choose a small positive constant ϵ ($\epsilon > 0$), which indicates the accuracy of the computation.
2) Find x_{upper} such that $\int_{x_{upper}}^\infty J(u)\ du \leq \epsilon$.
3) Let $B_0(x) = 1$ for all x and let $B_i(x) = 1, \quad \forall x > x_{upper}, \ \forall\ i$.
4) Also compute $K(x) = \int_x^\infty J(u)\ du$ for $A(x_{upper}) \leq x \leq x_{upper}$. Denote $K(A(x_{upper}))$ by ζ.
5) For all values of i, let $B_i(x) = 1 - K(x)$, for $A(x_{upper}) \leq x \leq x_{upper}$.
6) Repeat the following iteration in the range $(0, A(x_{upper}))$ until the function converges below a specified bound:
$$B_i(x) = 1 - \int_x^{A(x_{upper})} J(u)\ B_{i-1}(\beta u)\ du\ -\ \zeta.$$
7) Let the final solution be denoted by $B(x)$.
8) Renormalize $B(x) = \frac{B(x)}{B(0)}$ to satisfy the necessary boundary conditions. $B(x)$ is then the numerical estimate for $H(x)$.
9) The complementary probability distribution is then obtained as
$$G(x) = B(x)e^{-Q(x)} \tag{16}$$
10) Compute $F_{subj}(x)$ from $F_{subj}(x) = 1 - G(x)$.

V. Results for Classical Congestion Avoidance

We now compare the analytical results of the previous section with those obtained via simulations. The simulations were carried out with the TCP Reno and NewReno versions in the *ns-2* [22] simulator package. Although these versions differ in their fast recovery mechanisms and in the frequency of timeouts, the performance of the two versions was found to be almost identical for the relatively low loss environments studied in our simulations. To obtain adequate statistical confidence, simulation results were obtained by averaging over runs with multiple seeds; each run comprised at least 10^6 packet transmissions. While the entire simulation process would take $\sim 10 - 15$ minutes, the numerical computation over a fairly fine grid (~ 1000 points) took only about 30 secs (on a typical workstation).

A. TCP with Simple State-Dependent Loss

The results in Fig.1 correspond to the case when the packet drop probability depends directly on the window size. We

achieve this effect by passing a TCP connection through a single queue with negligible link propagation and transmission delay (all outstanding packets are thus effectively resident in the queue), and independently dropping each arriving packet with a probability that varies with the queue occupancy. The drop probability in this example increases linearly with queue occupancy. It can be seen that the simulated behavior offers excellent agreement with the numerical prediction in this example.

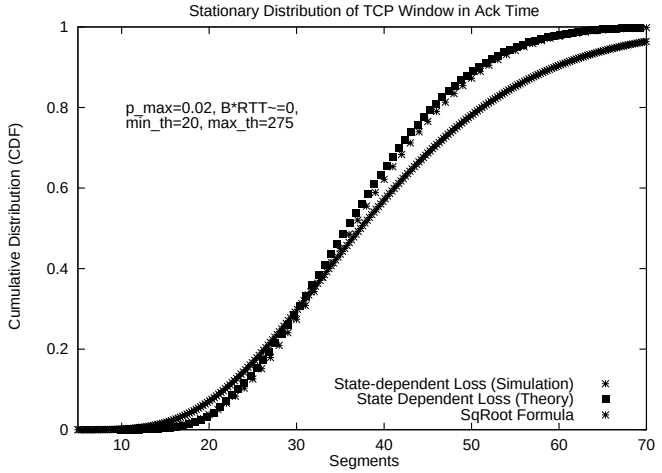

Figure 1: TCP Window Evolution and State-Dependent Loss

B. Predicting TCP behavior with Queue Management Techniques

One of the goals of our analysis is to predict the window distribution of a persistent TCP flow when it interacts with router queue management mechanisms like *Early Random Drop* (**ERD**) and *Random Early Detection* (**RED**), where the packet drop probability is not constant but varies with the queue occupancy. In the present paper, we consider the case where the router port buffers only a single flow; approximate techniques for determining the window distribution for multiple TCP flows were presented in [21].

While both ERD and RED involve variable drop probabilities that depend on the queue occupancy, they have significant differences, of which the two most important are:

- The drop probability in RED is dependent on an EWMA of the queue occupancy, while the drop probability in ERD is a function of the instantaneous queue length.
- RED uses *drop-biasing* to generate an inter-drop gap that is uniformly distributed; ERD drops each packet with an independent drop probability, resulting in inter-drop gaps that are geometrically distributed.

These differences make RED much harder to model than ERD: the use of averaged queue occupancies to determine drop probabilities destroys the state-dependent loss model (the drop probability is then a function of the past state behavior), while drop-biasing negates the assumption of independent packet drops. We circumvent these problems by (simplistically) assuming that the drop probability depends only on the instantaneous queue length and that each packet is dropped independently. We thus ignore the effect of queue averaging in RED; we include a simple correction to account for the effect of drop-biasing.

Assuming that the transmission pipe is always full, the occupancy of the queue is given by the residual number of unacknowledged packets, so that we have

$$Q_n = W_n - B.RTT \tag{17}$$

For our experiments, the loss function is given by the traditional model of RED behavior, i.e., $p(Q) = 0$ for $Q \leq min_{th}$, $p(Q) = p_{max}$ for $Q \geq max_{th}$ and $p(Q) = \frac{Q - min_{th}}{max_{th} - min_{th}} p_{max}$ for $min_{th} < Q < max_{th}$. The loss probability as a function of the window size is then given by $p(W - B.RTT)$

Illustrative results of our validation experiments are provided in figures 2 and 3, which plot the numerically predicted cumulative distribution of the TCP window against that obtained from simulations.

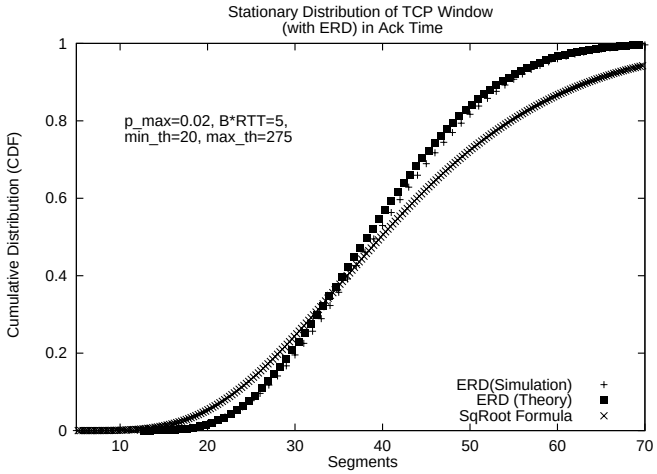

Figure 2: Behavior of TCP Window with Early Random Drop (and External Delay)

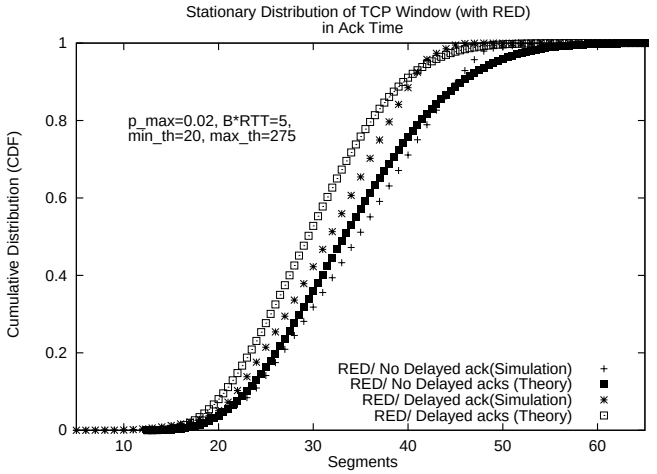

Figure 3: Behavior of TCP Window with Random Early Detection (with & without external delay and delayed acks)

C. Incorporating Delayed Acknowledgments

Our model of TCP window evolution has so far assumed that TCP receivers generate an acknowledgme nt for every arriving packet. Many implementations, however, use delayed acknowledgments to slow the rate of window expansion or alleviate congestion on the reverse link. We can model this artifact by noting that if the receiver sends one ack for every K packets received, then the TCP window grows from W to $W+\frac{1}{W}$ for every K packets transmitted. An approximation to this behavior is achieved by supposing that the TCP window grows by only $1/K^{th}$ of its value for every packet transmitted i.e., by modifying the window evolution equation to $W_{n+1} = W_n + \frac{1}{K.W_n}$.

Numerical results verify the effectiveness of this correction in accounting for the phenomenon of delayed acknowledgments. The graphs in figure 3 contain the comparisons between analysis and simulations when a TCP connection performing delayed acknowledgments is combined with the RED queue management algorithm, while figure 4 shows the comparisons when a TCP performing delayed acknowledgments interacts with the ERD queue management algorithm.

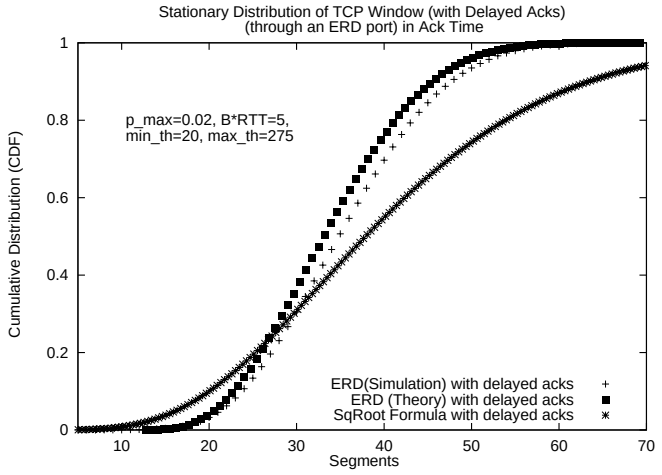

Figure 4: Behavior of TCP Window with Delayed Ack and Early Random Drop

VI. Modeling the Generalized Congestion Avoidance Algorithm

Having demonstrated the accuracy of our analysis for the case of classical congestion avoidance, we now extend the technique to analyze the generalized congestion avoidance algorithm.

By ignoring transients related to fast recovery and timeouts, the window evolution under generalized congestion avoidance is a Markov process with the following conditional probabilities:

$$P\{W_{n+1} = w + c_1 w^{\alpha} | W_n = w\} = 1 - p(w) \tag{18}$$

$$P\{W_{n+1} = w - c_2 w^{\beta} | W_n = w\} = p(w). \tag{19}$$

As in section III.B, we proceed by scaling the process $(W_n)_{n=1}^{\infty}$ in both the state-space and time axis. For the generalized case, we use the following state and subjective-time mappings:

$$X(t) = p_{max}^{\frac{1}{1-\alpha}} W_n \tag{20}$$

$$\Delta t = p(W_n)\Delta n \tag{21}$$

As in the classical congestion avoidance, the state-space rescaling is a constant, while the time-rescaling is state-dependent.

Theorem 3: *It can be shown, that as $p_{max} \downarrow 0$, the process $X(t)$, defined by equations (20) and (21) converges (path-wise) to a process whose window $X(t)$ behaves as follows: There is a Poisson process with intensity 1, with points denoted by $(\tau_n)_{n=1}^{\infty}$. In between the points of this Poisson process, X evolves according to the equation*

$$\frac{dX}{dt} = \frac{c_1 * p_{max} * X^{\alpha}}{p(\frac{X}{p_{max}^{\frac{1}{1-\alpha}}})}. \tag{22}$$

At the points of the realization of the Poisson process, we have

$$X(\tau^+) = X(\tau^-) * (1 - c_2)$$

Accordingly, we can now apply the elaborate numerical procedure presented in section IV to derive the stationary distribution of $X(t)$. After computing this stationary distribution, we simply reverse the space and time-scalings employed (as in section III.C) to obtain $F_{ack}(\ldots)$, the distribution of the generalized TCP window in ack time.

VII. Results for Generalized Congestion Avoidance

We now discuss a practical application of this generalized analysis. In particular, we determine the window distribution of a single generalized TCP flow under the Assured Service Model when it interacts with a single bottleneck queue. The Assured Service model [9] describes a framework for differential bandwidth sharing, where each flow (user) is guaranteed a minimum or *assured* rate as part of their service profile. Adequate capacity provisioning is assumed to ensure that packets from a flow experience minimal congestive losses/ marking as long as its transmission rate lies within this assured rate. Flows are allowed to inject additional (opportunistic) packets beyond this assured rate; such packets are treated as best-effort and have lower priority. To enable network buffers to differentiate between such packets, [9] proposes a tagging mechanism at the network edge. Packets which stay within the profiled rate are tagged as *in* packets while packets that violate the profile are tagged as *out* packets; mechanisms such as a leaky bucket [23] or modifications thereof [9] may be used to implement the tagging operation. *In* packets are provided preferential treatment in network buffers via the RIO (RED with In/Out) discard algorithm; RIO is similar to RED except that it uses different thresholds for *in* and *out* packets to ensure that *out* (opportunistic) packets were dropped before *in* packets. We assume that out bottleneck queue uses the ORED buffer management algorithm; ORED is similar to RIO but differs in two respects:

- ORED marks *out* packets instead of dropping them.
- ORED does not signal congestion notification for *in* packets, except when the buffer overflows and packets are dropped.

A. Mathematical Model

The persistent TCP is assumed to have a round-trip time of RTT secs and a maximum segment size (MSS) of M bytes. It interacts with an ORED buffer serving a link of capacity B MSSs/sec and is subject to an assured rate of R MSSs/sec. Our analysis assumes that

$$B > R. \tag{23}$$

The marking function of the ORED buffer (for *out* packets) is given by the traditional linear model: $f(Q) = 0$ for $Q \leq min_{th}$, $f(Q) = p_{max}$ for $Q \geq max_{th}$ and $f(Q) = \frac{Q - min_{th}}{max_{th} - min_{th}} p_{max}$ for $min_{th} < Q < max_{th}$, where min_{th} and max_{th} are expressed in MSSs. Let Q and W represent the ORED buffer occupancy and the TCP window size respectively.

If, as before, we assume that buffer underflow never occurs, it is clear that the TCP average transmission rate will be equal to the link capacity B. The probability of a packet being tagged by a conditioner at the edge, γ, is then independent of W and Q, and is simply given by the fraction by which the capacity exceeds the profiled rate

$$\gamma = \frac{B - R}{B} \tag{24}$$

Also, as before, our assumption of no buffer underflow (for the bottleneck queue) implies that

$$W = Q + B * RTT \tag{25}$$

Now consider the evolution of the TCP generalized window. It is easy to see that although packets will be tagged as *out* as soon as the TCP throughput exceeds R, they will not be marked (ECN bit set) until the window has expanded to ensure that the queue occupancy exceeds min_{th}; this, of course, occurs only after the throughput has reached the bottleneck bandwidth B and the window size has exceeded $B * RTT + min_{th}$. Accordingly, a reasonably accurate model of the marking probability $p(W)$, as a function of the window size W, is given by the equations

$$\begin{aligned} p(W) &= 0 \qquad\qquad for \;\; W < min_{th} + B.RTT, \\ &= \gamma * f(W - B.RTT) \; for \;\; W < max_{th} + B.RTT \\ &= \gamma * \bar{p_{max}} \qquad for \;\; W > max_{th} + B.RTT, \end{aligned} \tag{26}$$

where $\gamma = \frac{B-R}{B}$. Having obtained an expression for $p(W)$ in equations (18) and (19), we can then obtain the stationary window distribution of the TCP process using the mappings in section V.

B. Results

To illustrate the accuracy of our analysis, we take the classical congestion avoidance parameters ($\alpha = -1$, $\beta = 1$, $c_1 = 1$ and $c_2 = 0.5$) as a baseline parameter set and vary each of the three parameters α, c_1 and c_2 in turn. A set of typical results are provided here, for the following network parameters: an MSS of 512 bytes, nominal RTT of 13.66 msec, an assured rate of 0.75 Mbps and an ORED queue with a service rate of 3 Mbps (the bandwidth-delay product is thus 5 segments), $min_{th} = 15$, $max_{th} = 95$ and $p_{max} = 0.02$.

Figure 5 shows the simulated and theoretical mean and variance of the window size of the TCP flow as a function of α and attests to the accuracy of our analysis. We see that an increase in α not only increases the mean window size but also the the coefficient of variation (defined as $\frac{Std.Deviation(W)}{Mean(W)}$). Note also that our technique becomes less accurate as α increases. A larger α implies a larger mean queue occupancy and hence a larger average marking probability; accordingly, our mathematical approximation, which is clearly based on the limiting process as $p_{max} \downarrow 0$, will be progressively less applicable.

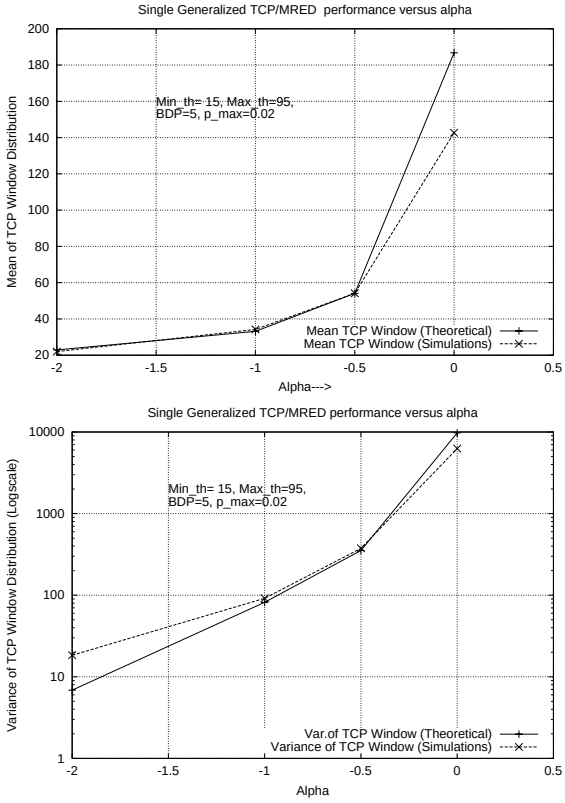

Figure 5: Statistics of Generalized TCP with α

We have also studied the window statistics and distribution by varying c_1 and verified the accuracy of our technique. The figures do not provide any great insight and are thus omitted here. Figure 6 shows the plots of the TCP window statistics when the decrease coefficient, c_2, is varied. We note that as c_2 is decreased from its current value of 0.5, the mean window size increases but the variance decreases, i.e., *the coefficient of variation decreases rapidly*. [5] contains an elaborate discussion on preferred changes in the parameter c_2 and shows how a higher value of c_2 (less aggressive decrease)

can be leveraged to provide better TCP dynamics in ECN-enabled environments.

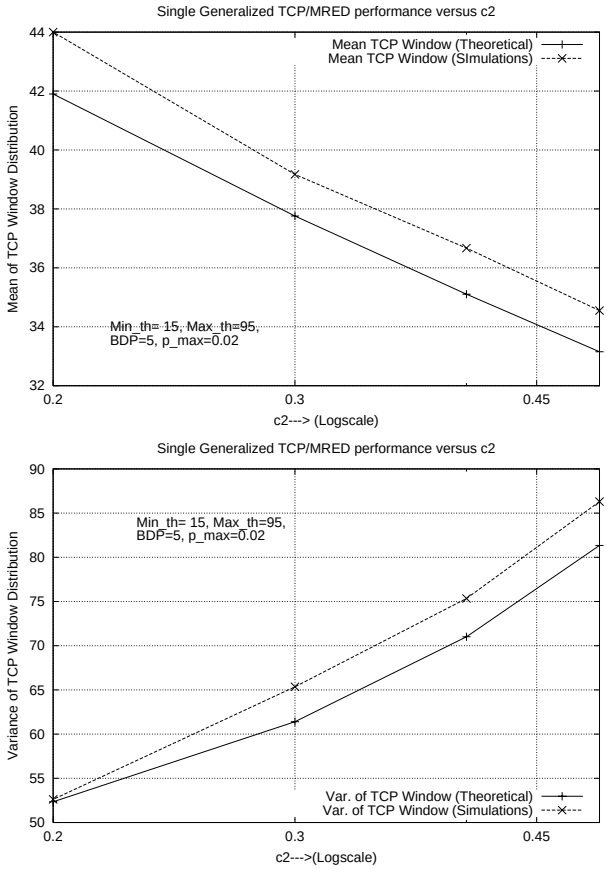

Figure 6: Statistics of Generalized TCP with c_2

VIII. Conclusions

In this paper, we presented a technique for analyzing and predicting the window distribution of a persistent TCP connection subject to randomized congestion notification with a variable, but *state-dependent*, notification probability. The main contribution of this paper is the state-dependent time-rescaling technique, which allows us to convert the discrete-time Markovian TCP process to a process that can be described by a Poisson-driven stochastic differential equation (SDE). This re-scaling technique is generic enough to be applied to any arbitrary continuous-space, continuous-time Markov process that is subject to a stationary failure process.

We first considered the case of classical TCP congestion avoidance and subsequently extend the technique to consider the broader class of generalized congestion avoidance algorithms, where for every incoming acknowledgment, the TCP flow increases its window by $c_1 W^{\alpha}$ in the absence of congestion and decreases its window by $c_2 W^{\beta}$ in the presence of congestion. By studying the process in subjective time (which is a history-dependent rescaling of the time index), we can describe its evolution using a Poisson-driven SDE. We have also presented a rapidly and provably convergent numerical technique for solving this SDE, as well as the space and time re-scalings needed to eventually obtain the stationary distribution of the original Markov process. Comparisons with simulation results suggest that this technique is fairly accurate in predicting the distribution and other statistics of the congestion window.

Further simulations involving generalized congestion avoidance have also demonstrate the accuracy and applicability of our technique under the Assured Service model. We have found that decreasing c_2 (which may be possible if ECN-capable routers provide stronger feedback) appears to be an attractive modification, since it appreciably lowers the coefficient of variation of the window size.

References

[1] V. Jacobson and M. Karels, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM'88, August 1988.
[2] T. Ott, M. Matthis and J. Kemperman, "The Stationary Behavior of Ideal Congestion Avoidance", http://web.njit.edu/õtt/Papers/Mathis/ TCP-window.pdf, August 1996.
[3] V. Jacobson, "Modified TCP congestion avoidance algorithm", April 30, 1990, end2end-interest mailing list.
[4] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms", Proceedings of IEEE INFOCOM, April 2001.
[5] A. Misra and T.Ott, "Jointly Coordinating ECN and TCP for Rapid Adaptation to Varying Bandwidth", Proceedings of IEEE MILCOM, October 2001.
[6] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993.
[7] K. K. Ramakrishnan and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999.
[8] A. Misra, J. Baras and T. Ott, "Generalized TCP Congestion Avoidance and Its Effect on Bandwidth Sharing and Variability", Proceedings of Globecom 2000, December 2000.
[9] D. Clark and W. Fang, "Explicit Allocation of Best Effort packet Delivery Service", IEEE/ACM Transactions on Networking, August, 1998.
[10] S. Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic", Computer Communication Review, Vol.21, No.5, October 1991.
[11] T. V. Lakshman, U. Madhow and B. Suter, "Window-based Error Recovery and Flow Control with a Slow Acknowledgment Channel: a Study of TCP/IP Performance", Proceedings of Infocom '97, April 1997.
[12] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", Proceedings of Sigcomm '98, September 1998.
[13] A. Kumar, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link", IEEE/ACM Transactions on Networking, August 1998.
[14] V. Misra, W. Gong and D. Towsley, "Stochastic Differential Equation Modeling and Analysis of TCP Windowsize Behavior", Proceedings of Performance'99, October 1999.
[15] V. Misra, W. Gong and D. Towsley, "A Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", Proceedings of ACM SIGCOMM'00, September 2000.
[16] F. Baccelli and D. Hong, "TCP is Max-Plus Linear", Proceedings of ACM SIGCOMM'00, September 2000.a
[17] E. Altman, K. Avrachenkov and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses", Proceedings of ACM SIGCOMM'00, September 2000.
[18] M. Matthis, J. Semke, J. Mahdavi and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communications Review, July 1997.
[19] E. Hashem, "Analysis of Random Drop for Gateway Congestion Control", MIT Technical Report, MIT-LCS-TR-506.
[20] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", IEEE/ACM Transactions on Networking, June 1993.
[21] A. Misra, T. Ott and J. Baras, "The Window Distribution of Multiple TCPs with Random Loss Queues", Proceedings of Globecom '99, December 1999.
[22] The ns-2 network simulator, http://www.isi.edu/nsnam/ns/.
[23] M. Schwartz, "Broadband Integrated Networks", Prentice Hall, 1997.
[24] T. Ott, "ECN Protocols and the TCP Paradigm", http://web.njit.edu/õtt/Papers/ECN/ECN.pdf.