

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

Archan Misra
archan@us.ibm.com

Teunis J Ott
ott@oak.njit.edu

John Baras
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.

Keywords—Markov processes, TCP, window distribution, rescaling, congestion avoidance, variable, loss, marking, RED, ECN.

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.

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

Archan Misra is with IBM T J Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532. Teunis Ott is with the Computer Science Department, New Jersey Institute of Technology, Newark, USA. John Baras is with the Center for Satellite and Hybrid Communication Networks at the University of Maryland, College Park, Maryland, MD 20742, USA.

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 packet¹ has arrived at the source. The resulting discrete-time Markov process exhibits the following conditional probabilities:

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

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

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,

¹In the case of packet drop, where congestion is indicated by the absence of an acknowledgment, we can conceptually assume the arrival of a ‘phantom’ negative acknowledgment.

however, assume a constant drop or marking probability. The ‘square-root’ formula, which states that the average window of a persistent TCP connection is 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.

Several recent research efforts have also explored the use of differential equations to model the dynamics of TCP window evolution. By treating the congestion notification events as a Poisson process with rate λ , [14] treats the window evolution of TCP through a stochastic differential equation (much like we do). This differential equation can then be solved to derive various moments of the window size distribution. Moreover, this approach can explicitly account for timeout events and restrictions on the maximum window size. Additionally, [15] extends this analytical technique based on Poisson drops to a closed system, where the Poisson rate is not constant, but rather, $\lambda(\rho)$, a function of the TCP instantaneous rate ρ . However, the Poisson loss assumption is justified only through a limiting assumption that holds when a flow traverses a very large number of queues, a situation unlikely to be observed in practice. In contrast, our analysis starts with an explicit state-dependent congestion notification model and introduces a *time-rescaling* that rigorously generates an equivalent Poisson driven Stochastic Differential Equation model, *that holds theoretically for any number of queues*. Another interesting and mathematically rigorous approach for analyzing TCP dynamics employs the use of ‘max-plus’ algebra to study the dynamic evolution of discrete-time processes. This approach was used, for example in [16], to study the dynamical behavior of TCP Tahoe and Reno through a linear dynamical system, and derive formulas for TCP throughput in the presence of multiple bottlenecks. However, the algebra holds when losses (in general, congestion feedback events) are generated either deterministically (tail-drop buffers), or occur as pure i.i.d. events (no correlation to the flow’s window size). Finally, [17] analyzes the window dynamics of TCP under the assumption of a stationary, but otherwise general, feedback process. (In contrast, our analysis is more restrictive, since it applies only to independent feedback events). While [17] proves that the window distribution has a stationary solution that can be expressed as an infinite sum of independent random variables, it concentrates on deriving only the first two moments, rather than the

entire distribution.

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. We shall demonstrate fairly good agreement with simulation results for both the ERD and, surprisingly enough, the RED case, even though RED's use of an exponentially-weighted moving average (EWMA) for the queue occupancy violates our assumption of memoryless loss events.

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.

A few words are in order about the applicability of this analysis to TCP behavior. Our Markovian model of process evolution ignores transients such as timeouts and fast recovery: both [12] and [13], on the other hand, show the importance of timeouts when packet loss rates are high. When congestion notification is achieved via packet drops, our analysis is accurate as long as the loss probabilities are relatively small (less than $\sim 1 - 5\%$) and the delay-bandwidth product (including the buffering delay) high enough (≈ 10 MSSs and above) to ensure that timeouts are relatively rare events. When ECN is used to explicitly signify congestion, we can safely ignore timeout-related transients since packet losses are absent and treat the TCP window evolution as a Markovian process. However, our mathematical technique is based on a limiting analysis (we make $p \downarrow 0$); accordingly, our numerical results are more accurate when the *marking* probabilities are not excessively large. The disproportionate impact of timeouts in current TCP versions is due to the combined effects of coarse-grained timers and the integration of loss recovery mechanisms with congestion control in current TCP versions. When loss recovery is separated from adaptation to congestion (as in SACK TCP), timeouts will begin to play a relatively less important role and the range of *loss* probabilities over which this model holds for TCP behavior will increase.

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 character-

ized by Poisson points of failure. After proving the properties of interest of this continuous time process, we show how a more general Markov process with state-dependent transition probabilities can similarly be rescaled to one that is characterized by Poisson points of failure, and is thus amenable to analysis via a Kolmogorov equation.

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:

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

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

where p is the packet dropping probability². Our Markovian formulation holds when, given the current window size, the generation of congestion notification is conditionally independent of past and future congestion events. As in ([2]), we will approximate this process by a more amenable continuous-space, continuous-time process.

The time index in equations (3) & (4) is called *ack time* and 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 $F_{ack}(\cdot)$.

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 $dt_{subjective} = p \cdot dt_{ack}$. When the loss probability is not constant but state-dependent, a state-dependent (non-linear) scaling must be used.

²Unless otherwise stated, we shall use the terms 'packet dropping' and 'packet marking' interchangeably, since they are equivalent indicators of congestion, as far as this analysis is concerned.

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

$$\Delta t = p(W_n)\Delta n \quad (5)$$

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

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

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 $\frac{p(W)}{p_{max}} > \epsilon \forall W$ ³, (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

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

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

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

Equation (5) implies that a loss probability of zero ($p(W_n) = 0$) results in an increase in ack time but no increase in subjective time. Subjective time thus loses information about the process behavior during those ack times when the system evolves deterministically without loss; the mapping in equation (5) is non-invertible if $p(W_n)$ is 0. Later in the paper, we shall see how to

³The above requirement may, in several cases, be more stringent than practically necessary. For example, when the variation of loss probability with window size is very gradual, the bulk of the distribution mass will often lie around some small value of p , say p^* . We can then use $\sqrt{p^*}$ as our space-rescaling factor; for the rescaled process to be well-behaved, we then merely need $\frac{p(W)}{p^*}$ to be bounded away from 0.

correct the stationary distribution of the process for portions of the state-space where the loss probability equals 0; for the time being, assume that $p(W_n) \neq 0$ in the region of interest.

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

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

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

Proof:

The proof of the differential equation describing the window evolution between failure events is trivial and obtained by taking appropriate limits in equations (3), (5) & (6). The relationship at an instant of failure also follows easily from equations (4) and (6). Note that the derivative in equation (8) is always well-defined by virtue of our assumption that $p(W_N) > 0$ for the interval under consideration. The proof that the instants of failure become a realization of a Poisson process of intensity 1 is provided in Appendix I. It consists of showing that as $p_{max} \downarrow 0$, the number of packet transmission events in any finite interval T becomes infinitely large and the probability of loss of each transmitted packet is such that $Prob(\text{no loss in an interval } T) = e^{-T}$. \diamond

The process defined in Theorem 1 is an approximation of the re-scaled ‘TCP’ process; the approximation becomes asymptotically accurate as the loss probabilities become smaller. For a given loss probability function $p(W)$, we analyze the rescaled ‘TCP’ process by assuming that it exhibits the behavior of the limiting process outlined by Theorem 1. In other words, even for a finite loss probability, we assume that $W(t)$ is described by the differential equation (8), with an i.i.d and exponential distribution of times between packet drops. We can thus expect the numerical analysis outlined later to predict TCP window behavior more accurately as p_{max} becomes smaller.

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 III. 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 $F_s(w) = F_{subj}(\sqrt{p_{max}}w)$.

Our desired distribution $F_{ack}(w)$ can then be obtained by noting that the state-dependent rescaling of subjective time (in

equation (5)) introduces a sampling non-uniformity in the process $W(t)$. To see this non-uniformity, note that an acknowledgment arriving when the window is w occupies an interval of 1 in ack time but corresponds to an interval $p(w)$ in subjective time: a uniformly distributed sampling on the subjective time axis corresponds to a non-uniform sampling (with non-uniformity proportional to $p(w)$) in the ack time frame. As long as the process $W(t)$ is ergodic, this sampling bias corresponds to a completely equivalent non-uniformity in the stationary distribution.

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

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

D. A Generalized Process

The analysis used to derive the stationary distribution of $W(t)$ is applicable to a more general class of processes. We now present a generalized notion of subjective time by considering a continuous-time stochastic process, $X(t)$, with a state-dependent failure rate $\lambda(x)$. We can now derive another process $Y(\tau)$ from $X(t)$ such that an increase of dt in the time index t of $X(t)$ corresponds to an increment of $\lambda(X(t))dt$ in the time index τ of $Y(\tau)$. A realization of the process Y will thus assume the same state-space values as the corresponding realization of X but at *different instants of time*. Subjective time can also be thought of as a *history-and-state dependent rescaling* of the base (ack) time index⁴ **The importance of the process $Y(\tau)$ lies in the fact that $Y(\tau)$ will now have a constant failure rate in its own notion of time (proved in Appendix I).** The time index, τ , of the process $Y(\tau)$ is then known as *subjective time* in reference to the time index t of the process $X(t)$ and the two are related by the differential relation

$$d\tau = \lambda(X(t))dt \quad (10)$$

Subjective time can also be considered to be a variable stretching (or contraction) of the time index.

We can thus see that any arbitrary process with a state-dependent failure rate can be reduced to a process with a constant Poisson failure rate by moving to an appropriate subjective time. Thus, we do not lose generality by considering only processes with constant failure rates. Accordingly, we can then consider a general process $W(t)$, described by the differential equation

$$\frac{dW}{dt} = \frac{1}{q(W)} \quad (11)$$

in between the instants of failure of a Poisson process with rate λ ; let q be a well-behaved function (finitely many discontinuities) such that $q(w) > 0 \forall w$. At the instants of failure of

⁴Readers familiar with Weighted Fair Queuing (WFQ) ([20]) may benefit from realizing that our subjective time formulation is analogous to that the definition of *virtual time* in WFQ; both attempt a state-dependent rescaling of time so that the process of interest has an invariant behavior in the new time scale.

the Poisson process, the process evolution is given by $W(t^+) = A(W(t^-))$, where $A(w) : [0, \infty) \rightarrow [0, \infty)$ is a strictly increasing function of w such that $A(w) < w, \forall w > 0, A(0) = 0$. Since A is strictly increasing, it has an inverse function $a(w)$, such that $a(A(w)) = w$ and $a(w) > w, \forall w > 0$. For the TCP-specific case at hand, we have $A(w) = \frac{1}{2}w$ (so that $a(x) = 2x$), the intensity λ of the Poisson process is 1 and the rate function $q(W) = \frac{p(\frac{W}{\sqrt{p_{max}}})W}{p_{max}}$.

In the next section, we shall formulate and solve the Kolmogorov equation for this generalized process $W(t)$.

IV. THE STATIONARY KOLMOGOROV EQUATION AND ITS SOLUTION

In this section we obtain the stationary distribution of the process, defined in section III.D, 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.D satisfies the differential equation

$$\frac{dF_{subj}(x)}{dx} = \lambda q(x)(F_{subj}(a(x)) - F_{subj}(x)) \quad (12)$$

Proof:

If $F_{subj}(x, t)$ is the cumulative distribution function at (subjective) time t , then the distributions at times t and $t + \Delta t$ can be related as

$$F_{subj}(x + \frac{\Delta t}{q(x)}, t + \Delta t) =$$

$$F_{subj}(x, t) + \lambda \Delta t (F_{subj}(a(x)) - F_{subj}(x))$$

The first term in the RHS of the above equation asserts that the process cannot increase by more than $\frac{\Delta t}{q(x)}$ in an interval of time Δt while the second term considers the probability of loss events that would cause the process value to reduce below x at time $t + \Delta t$. Since the stationary distribution $F_{subj}(x)$ is invariant in t , we get the resulting differential equation

$$\frac{dF_{subj}(x)}{dx} = \lambda q(x)(F_{subj}(a(x)) - F_{subj}(x)) \quad (13)$$

◇

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

$$\frac{dF_{subj}(x)}{dx} = q(x)(F_{subj}(2x) - F_{subj}(x)), \quad (14)$$

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 (12) is equivalent to the equation

$$\frac{dG(x)}{dx} + \lambda q(x)G(x) = \lambda q(x)G(a(x)) \quad (15)$$

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

$$H(x) = H(z) - \lambda \int_x^z q(u)e^{Q(u)}G(a(u))du \quad (16)$$

Now, suppose that $\lim_{x \uparrow \infty} H(x)$ exists and is equal to \bar{H} . \bar{H} will exist only if the tail of the complementary distribution decays as $e^{-Q(x)}$. By evaluating the behavior of equation (15) for very large x (where $G(a(x))$ can be considered to be 0 with negligible error), we can easily see that this phenomenon of exponential decay is indeed true. Now, by letting $z \uparrow \infty$ in equation (16) and noting that $G(a(u)) = e^{-Q(a(u))}H(a(u))$, we have

$$H(x) = \bar{H} - \lambda \int_x^\infty q(u)e^{(Q(u)-Q(a(u)))}H(a(u))du \quad (17)$$

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 (17) reduces to

$$H(x) = \bar{H} - \int_x^\infty J(u)H(a(u))du \quad (18)$$

By iterated expansion, $H(x)$ can be shown to obey the relation

$$H(x) = \bar{H} \sum_{k=0}^{\infty} (-1)^k \overbrace{\int_{u_1 > x} \dots \int_{u_k > \beta u_{k-1}}}^{k\text{-fold}} J(u_1) \dots J(u_k) du_k \dots du_1 \quad (19)$$

Appendix III provides a proof that the above infinite sum indeed converges to a limit when the function $q(x)$ is non-decreasing in x ; this condition holds for the TCP process whenever *the drop probability is a non-decreasing function of the window size*.

B. Numerical Computation

Repeated substitution in equation (18) 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 $B_i(x) = 1, \forall x > x_{upper}, \forall i$.
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

$$G(x) = B(x)e^{-Q(x)} \quad (20)$$

10. Compute $F_{subj}(x)$ from $F_{subj}(x) = 1 - G(x)$.

C. Correcting for Lossless Evolution

As noted in section II.A, the rescaled TCP process in subjective time cannot capture the dynamics of the window evolution when the loss probability is 0 (as subjective time freezes during these epochs). From a sample path point of view, the infinite derivative in equation (8) (Proposition 1) and the zero time increment in equation (5) imply that whenever the TCP process (in subjective time) enters an interval in the state-space corresponding to 0 loss, it instantaneously jumps from the lower to the upper end of the interval. In this subsection, we show how $F_{ack}(x)$ for the TCP process, obtained from the mapping in equation (9), can be corrected to incorporate the dynamics of the lossless evolution; the corresponding correction for the generalized process is then straightforward.

The correction for the density $f_{ack}(x)$ in ack time (after the correction for state-space rescaling has been completed) is computed by the *level crossing principle* which equates the rate at which the process evolves to the right of a value x to the rate at which the process transitions to the left. For the TCP process, this results in the equality

$$f_{ack}(x) \frac{1}{x} = \int_x^{2x} p(u) dF_{ack}(u). \quad (21)$$

This follows by noting that at a point x , TCP evolves to the right at a rate $\frac{1}{x}$ while it moves to the left at the rate governed by the loss rate in the interval $(x, 2x)$. By first obtaining the values of $F_{ack}(x)$ (up to a scaling constant) in the regions with non-zero loss probability, we can correct the solution for regions

with zero loss probability using the equation (21)⁵ (If $F_{ack}(u)$ in the RHS of equation (21) is unknown for any u , it follows that $p(u) = 0$ also; the unknown region may thus be left out of the computation.)

The numerical recipe for correcting the distribution for the lossless region is:

1. For the region(s) where $p(x) = 0$, compute the density $f_{ack}(x)$ using the level crossing relation

$$f_{ack}(x) = x \int_x^{2x} p(u) f_{ack}(u) du \quad (22)$$

2. Renormalize $f_{ack}(x)$ by $\int f_{ack}(u) du$ over the entire state-space to ensure a well formed probability distribution function $f_{ack}(x)$.

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. For comparison purposes, we include the distribution predicted by the ‘square-root formula’ in (2) assuming a constant drop probability; the constant value of the drop probability was taken to be the drop probability corresponding to the mean TCP window size obtained via simulation. As expected, the ‘square-root’

⁵The level-crossing equation (21) is actually valid for the entire range of the TCP process state-space (and not just where $p(x)$ is 0). It can easily be seen that the equation (14) (in subjective time) is equivalent to equation (21) (in ack time) by noting that the following set of relations: $p(x)dF_{ack}(x) = K.dF_s(x)$; $f_{ack}(x) = \frac{K.f_s(x)}{p(x)}$; $F_s(x) = F_{subj}(\sqrt{p_{max}}x)$; and $f_s(x) = \sqrt{p_{max}}f_{subj}(\sqrt{p_{max}}x)$, where K is a normalizing constant. The elaborate rescalings and computations in subjective time in this paper are necessary simply because there does not appear to be a simple way of solving equation (21) directly over the entire state space! Another way of looking at the subjective time formulation is therefore to think of it merely as a change of measure that replaces equation (21) with the more tractable version in equation (14).

approximation predicts a much larger variation in the window size than the true distribution.

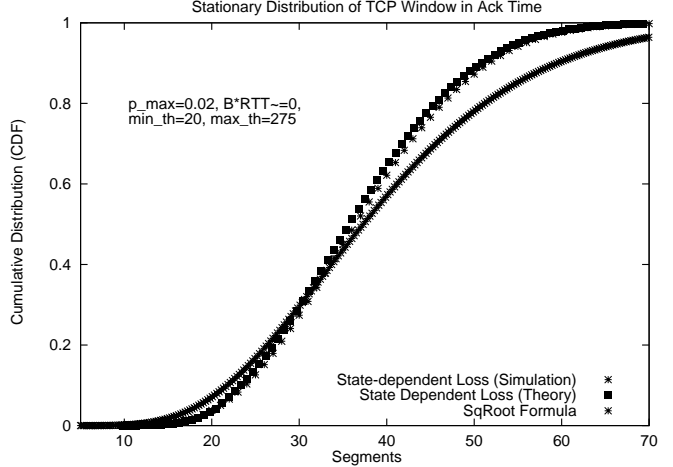


Figure 1: TCP Window Evolution and State-Dependent Loss

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 (discussed in Appendix II), 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 shall however present a simple correction to account for the effect of drop-biasing.

B.1 Relating the Loss Probability to Queue Occupancy

As already stated, we assume that the loss probability is determined by the instantaneous queue occupancy (for both RED and ERD); the loss probability for a given TCP window is derived by relating the queue occupancy to the TCP window. Neglecting the periods of fast recovery, the number of unacknowledged packets in flight, when the window is W_n , equals $\lfloor W_n \rfloor$, or in an approximate sense, W_n . If B (pkts/sec) is the service rate of the (bottleneck) queue and the round-trip delay (ignoring the queuing delay) is RTT (sec), then $B \cdot RTT$ packets are necessary to fill the transmission pipe. Assuming that this pipe is always full⁶, the occupancy of the queue is given by the residual number of unacknowledged packets, so that we have

$$Q_n = W_n - B \cdot RTT \quad (23)$$

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 $p(W - B \cdot RTT)$ ⁷

While the above model cannot capture the queue averaging function of RED, we can make a simple correction to approximate the effect of drop biasing in our model. We note that for a given value of drop probability p , the uniform distribution of inter-drop gaps in RED implies that the average gap is $\frac{1}{2p}$; the geometric distribution of gaps (resulting from an independent loss model) implies an average gap of $\frac{1}{p}$. For the RED simulations, we accordingly modify our analytical drop function such that our average agrees with that of RED, i.e., for a given queue occupancy q , we make $p_{model}(q) = 2p_{red}(q)$.

B.2 Experimental Results

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 shows that our numerical analysis provides an excellent match with simulation when the queue implements the ERD algorithm. The distribution predicted by the square-root formula is also provided for comparison. Figure 3 consists of two graphs, the top one for a RED queue with $B \cdot RTT \approx 0$ and the bottom one with $B \cdot RTT = 5$. The top graph isolates the effect of approximating the RED averaging process from the performance obtained when this approximation is combined with the assumption of a full pipe (equation (23)). The two graphs show, somewhat surprisingly, that the numerical predictions (with the correction for drop biasing) provide fairly close agreement with the simulated distribution when the queue implements RED. The closeness of the fit is somewhat

⁶This assumption holds only if the buffer never underflows (which, in turn, can hold only if the time taken by the buffer to drain \min_{th} packets is longer than RTT).

⁷Note that the ack arriving at the source at time n (when the window is W_n) corresponds to a packet generated a round-trip time earlier when the window was $W_{n'}$; the loss probability of the packet acked at n should thus be $p(W_{n'})$. However, as $cwnd$ increases by a maximum of 1 segment in a round-trip time $W_{n'} \approx W_n$, so that the loss probability of the packet acked at n can be assumed to be $p(W_n)$ with negligible error.

unexpected since the averaging effect in RED queues typically last over 500 packets; we expected this memory to significantly degrade the accuracy of our modeling.

C. Incorporating Delayed Acknowledgments

Our model of TCP window evolution has so far assumed that TCP receivers generate an acknowledgment 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 $W_{n+1} = W_n + \frac{1}{K \cdot W_n}$.

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. For the ERD queue, we also provide the theoretical distribution obtained by applying the correction for delayed acknowledgments in the square-root formula [2].

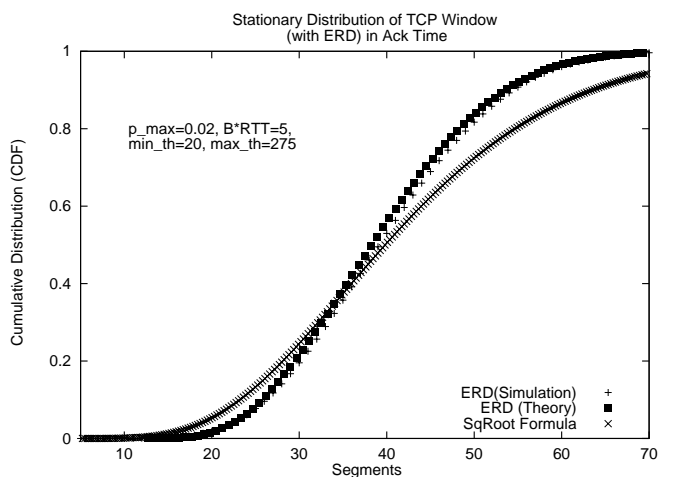


Figure 2: Behavior of TCP Window with Early Random Drop (and External Delay)

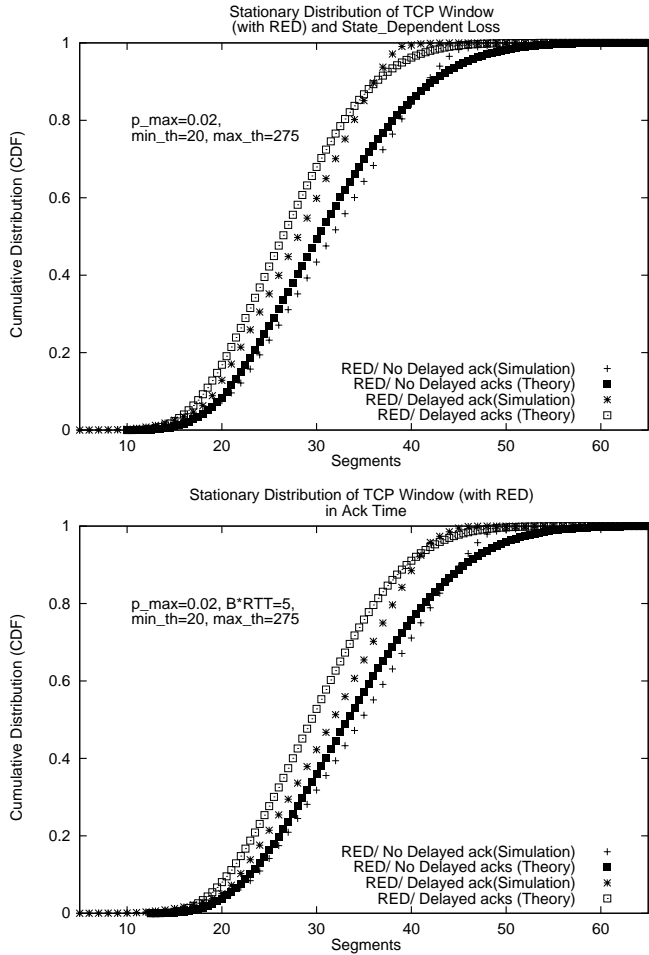


Figure 3: Behavior of TCP Window with Random Early Detection (with & without external delay and delayed acks)

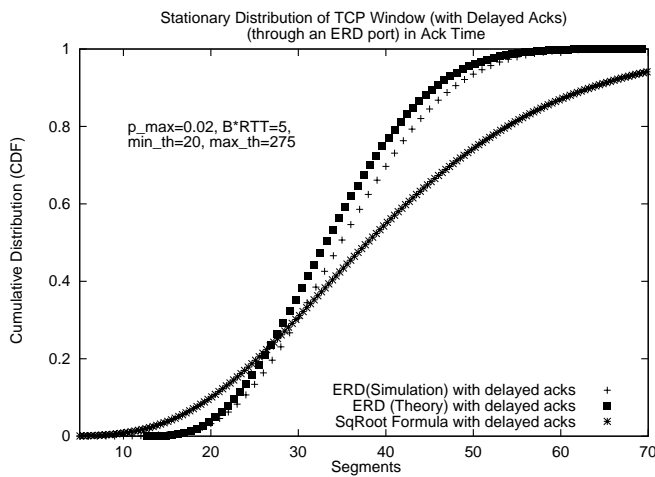


Figure 4: Behavior of TCP Window with Delayed Ack and Early Random Drop

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. Such an analysis will help us to study the implications of changes in TCP's current congestion avoidance algorithm. We shall shortly see that our extended technique is applicable only when $\beta = 0$. While a theoretical constraint, this condition is practically not very important since all suggested modifications to TCP congestion avoidance advocate *multiplicative-decrease* ($\beta = 0$).

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:

$$P\{W_{n+1} = w + c_1 w^\alpha | W_n = w\} = 1 - p(w) \quad (24)$$

$$P\{W_{n+1} = w - c_2 w^\beta | W_n = w\} = p(w). \quad (25)$$

As in section II.A, 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:

$$X(t) = p_{max}^{\frac{1}{1-\alpha}} W_n \quad (26)$$

$$\Delta t = p(W_n) \Delta n \quad (27)$$

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

Proposition 1: It can be shown (using arguments similar to Proposition 1), that as $p_{max} \downarrow 0$, the process $X(t)$, defined by equations (26) and (27) 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

$$\frac{dX}{dt} = \frac{c_1 * p_{max} * X^\alpha}{p\left(\frac{X}{p_{max}}\right)}. \quad (28)$$

At the points of the realization of the Poisson process⁸, we have

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

◇

Let $q(X)$ denote the inverse of the RHS of equation (8). It is then easy to see that the process $X(t)$ defined by Proposition 3 is identical to the process $W(t)$, presented in section II.D, under the following mappings:

$$q(W) = q(X)$$

$$A(W) = (1 - c_2) * W$$

⁸It is at a point τ of the Poisson process that the condition $\beta = 1$ is required. If $\beta \neq 1$, then $X(\tau^+)$ becomes ill-defined as p_{max} (and by implication, $p(\cdot)$) tends to 0.

Accordingly, we can now apply the elaborate numerical procedure presented in section III 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 II.B) to obtain $F_{ack}(\dots)$, 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 that⁹

$$B > R. \quad (29)$$

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 $f(Q) = \frac{Q - min_{th}}{max_{th} - min_{th}} p_{max}$

⁹If $B < R$, then ECN marking will occur even though the TCP flow obtains less than its assured rate. This is clearly a violation of the Assured Service model.

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

$$\gamma = \frac{B - R}{B} \quad (30)$$

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

$$W = Q + B * RTT \quad (31)$$

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

$$\begin{aligned} p(W) &= 0 && \text{for } W < min_{th} + B.RTT, \\ &= \gamma * f(W - B.RTT) && \text{for } W < max_{th} + B.RTT \\ &= \gamma * p_{max} && \text{for } W > max_{th} + B.RTT, \end{aligned} \quad (32)$$

where $\gamma = \frac{B-R}{B}$. Having obtained an expression for $p(W)$ in equations (24) and (25), we can then obtain the stationary window distribution of the TCP process using the mappings in section 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. To further demonstrate the accuracy of our numerical technique, we also include a plot comparing the predicted and simulated window *distribution* for $\alpha = -1.0$. 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.

We have similar 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. In general, we find that increasing c_1 increases not just the mean but the coefficient of variation as well. ([24] showed that the coefficient of variation would ideally be independent of c_1 if the marking probability was constant.)

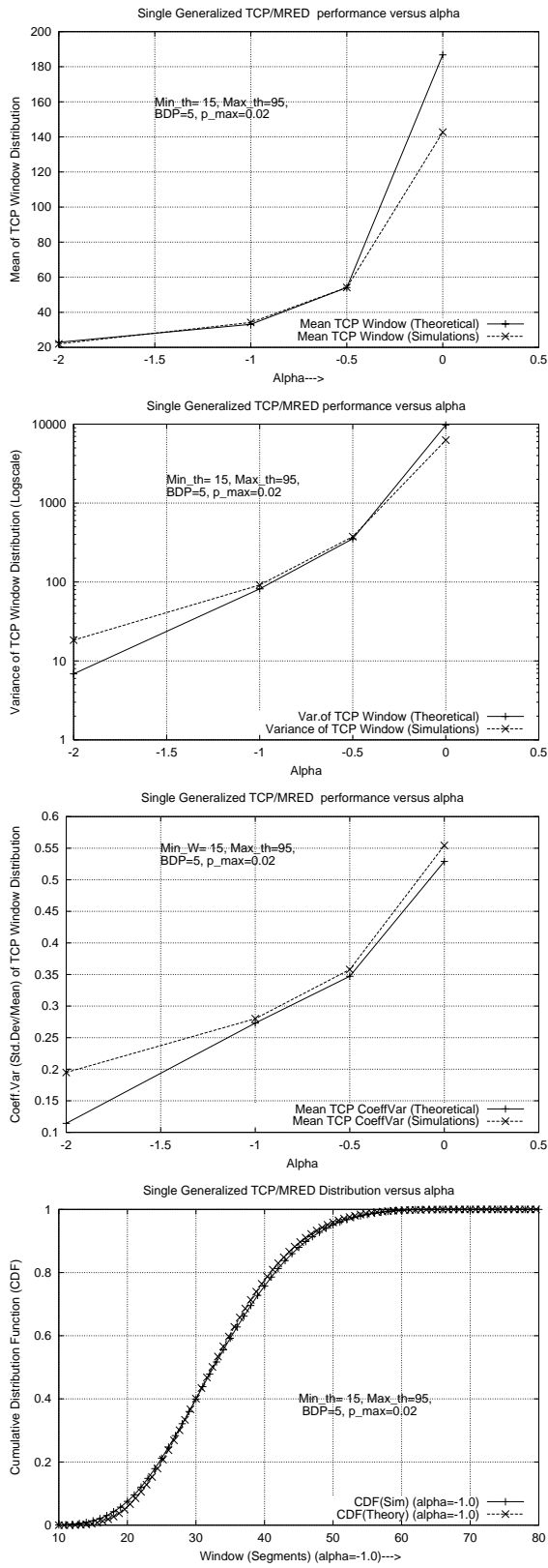
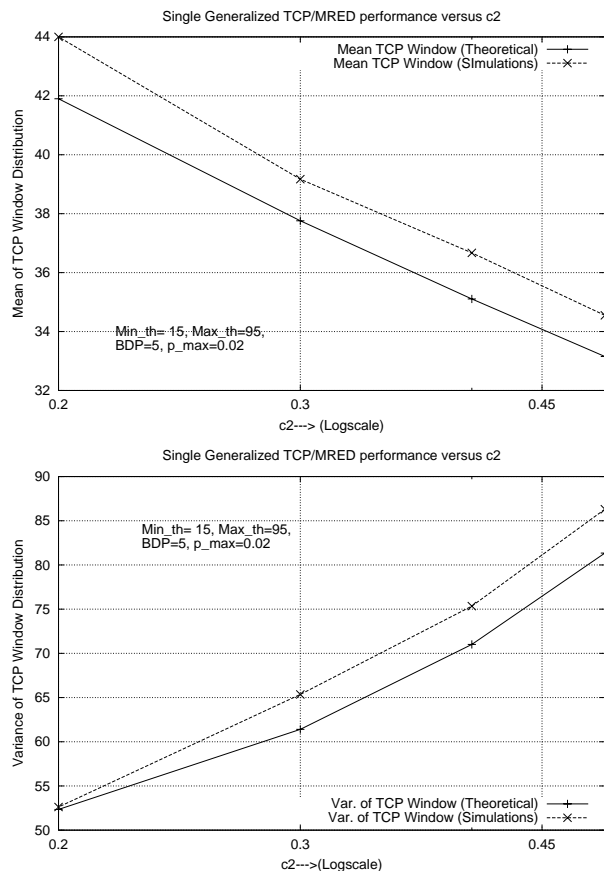


Figure 5: Distribution (and Statistics) of Generalized TCP with α

Figure 6 shows the plots of the TCP window statistics (as well as the simulated and theoretical distributions for $c_2 = 0.2$ and $c_2 = 0.4$) when the decrease coefficient, c_2 , is varied. (Note that [24] showed that the coefficient of variation is proportional to $\sqrt{\frac{1}{1-c_2}}$, when the marking probability is constant.) The figures clearly indicate the accuracy of our technique for computing the window distribution for various values of the generalized congestion avoidance parameter set. We also 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.



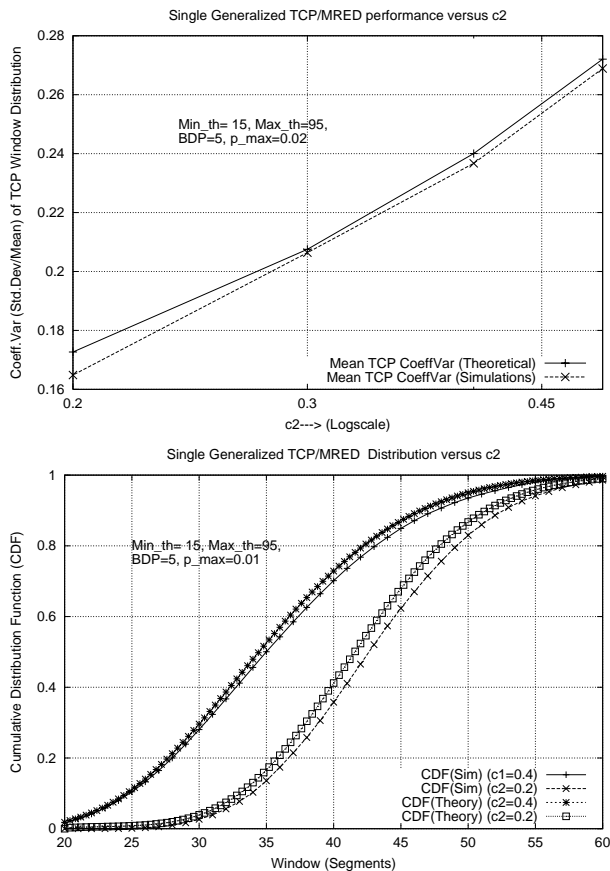


Figure 6: Distribution (and Statistics) of Generalized TCP with c_2

VIII. CONCLUSION

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 rescaling 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. Moreover, the numerical technique takes $O(\text{secs})$ to converge on a general purpose computer, as opposed to the $O(\text{mins})$ needed to obtain simulation-based results with acceptable confidence intervals. In particular, we find that this analysis can be used to predict the window behavior of a single persistent TCP flow interacting with buffer management algorithms such as ERD and RED. While the accuracy of the predictions was expected for ERD, the fit for the case of RED was surprisingly good. When congestion notification occurs via randomized packet *drops*, the analysis is more accurate when the loss probabilities are low and timeouts are relatively rare events.

Further simulations involving generalized congestion avoidance have also demonstrated the accuracy and applicability of our technique under the Assured Service model, where the TCP flow is provided a minimum bandwidth guarantee. We can thus use this analytical technique to study how changes in the algorithmic parameters affect the window distribution. 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. This observation was also reported using alternative studies and analyses in [5].

[21] combines the analysis presented here with a fixed-point based computation of mean window sizes to approximately evaluate the congestion window distribution in the presence of multiple TCP flows. Since this numerical technique is much faster than simulation-based studies, it appears promising as a technique to model TCP window behavior with much lower overhead, especially in hybrid and large-scale simulations. Moreover, the time-rescaling technique appears to be a powerful tool that can be applied to the study of Markov processes across a large number of disciplines.

APPENDICES

I. POISSON NATURE OF PACKET DROP EVENTS

We prove here that the subjective time formulation results in an inter-loss interval that is exponentially distributed with mean 1 and is independent of past and future intervals. For the TCP process under consideration, this property is asymptotically true as the loss/marking probabilities tend towards 0.

Let X_i be the random variable denoting the subjective time interval between the $(i-1)^{th}$ and i^{th} packet loss. Let us find the probability $P\{X_i > T\}$, i.e., the probability at least subjective time T elapses between the $(i-1)^{th}$ packet loss and the i^{th} packet loss. We renumber the packets: packet j , $j = (1, 2, \dots)$ denotes the j^{th} packet after the one that corresponds to the i^{th} loss. Since the congestion window is increasing after the i^{th} loss, there exists with probability 1 a (random!) N such that

$$p_1 + p_2 + \dots + p_N < T \leq p_1 + p_2 + \dots + p_{N+1}.$$

The probabilities p_j are also random.

The probability of interest is that none of the first N packets are lost. Since N is random, this probability equals

$$\sum_{n=1}^{\infty} P\{N = n\} \prod_{j=1}^n (1 - p_j). \quad (\text{I.1})$$

As long as (with probability one) $\max\{p_j, 1 \leq j \leq n\}$ is almost zero, the expression (I.1) is close to

$$\sum_{n=1}^{\infty} P\{N = n\} e^{-\sum_{j=1}^n p_j}. \quad (\text{I.2})$$

Since $0 < T - \sum_{j=1}^N p_j \leq p_{N+1}$, we see that as long as

$$\max\{p_j, 1 \leq j \leq N + 1\} \downarrow 0, \quad (\text{I.3})$$

the RHS of equation (I.2) equals

$$e^{-T} * \sum_{n=1}^{\infty} P\{N = n\} = e^{-T}. \quad (\text{I.4})$$

Hence, $P\{X_i > T\} \rightarrow e^{-T}$.

Since the above proof is also independent of the size of the packet that caused the i^{th} packet loss, we see that if condition (I.3) holds¹⁰, the inter-loss intervals (in subjective time) are not only exponentially distributed, but also independent of past and future intervals. This establishes the fact that the loss events are realizations of a Poisson process of intensity 1 in subjective time. \diamond

II. DIFFERENCES BETWEEN ERD AND RED

In this appendix, we discuss the differences between the Early Random Drop (ERD) and the Random Early Detection (RED) algorithms, which are important in understanding the applicability of our loss model. The important differences are:

- RED operates on the average (and not the instantaneous) queue length. The drop probability, p , is thus a function of the weighted average (Q_{avg}) of the queue occupancy i.e., p is a function not just of Q_n but of $(Q_n, Q_{n-1}, Q_{n-2}, \dots)$ with an exponential decay. Q_{avg} closely mirrors the instantaneous occupancy only if the queue varies slowly.
- To prevent large inter-drop durations, RED increases the drop probability for every accepted packet. (This property, which we call *drop-biasing*, is achieved by using a variable, cnt , which increases with every successive accepted packet; the true dropping probability is then given by $\frac{p(Q)}{1 - cnt \cdot p(Q)}$. This results in an inter-drop period that is uniformly distributed between $(1, \dots, \lfloor \frac{1}{p(Q)} \rfloor)$, as opposed to the independent drop model in ERD which results in geometrically distributed inter-drop periods.
- Some RED implementations have a sharp discontinuity in drop probability: when the average queue exceeds max_{th} ,

¹⁰Note that the condition (I.3) may hold even if p_{max} is not small. All we really need is the congestion window *almost always* stays small enough for $p(Q)$ to be small.

$p(Q)$ becomes 1 so that all incoming packets are deterministically dropped. This contrasts with our assumption that random drop occurs throughout the entire range of the buffer occupancy. Our analysis applies to such RED queues only if the TCP process almost never builds up queues that exceed max_{th} .

III. PROOF OF CONVERGENCE OF $H(x)$

To see that $H(x)$ in equation (19) indeed converges to a limit, let us define $C(x)$ by $C(x) = \int_x^{\infty} J(u) du$. Now, assume that there exists a $\beta > 1$, such that $A(x) \leq \frac{x}{\beta} \forall x$ (i.e., $a(x) \geq \beta x$). This is a stronger requirement than $A(x) < x$; in the case of the TCP model, $\beta = 2$. Now since $q(u)$ is a non-decreasing function of u ,

$$\int_u^{a(u)} q(\rho) d\rho \geq \int_u^{\beta u} q(\rho) d\rho \geq \frac{\beta u - u}{\beta u} \int_0^{\beta u} q(\rho) d\rho$$

so that

$$\int_u^{a(u)} q(\rho) d\rho \geq \frac{\beta - 1}{\beta} \int_0^{\beta u} q(\rho) d\rho \quad (\text{III.5})$$

Hence, $C(x) \leq \lambda \int_x^{\infty} q(u) e^{-\gamma Q(\beta u)} du$ where $\gamma = \frac{\beta - 1}{\beta}$. Thus,

$$C(x) \leq \lambda \int_x^{\infty} q(\beta u) e^{-\gamma Q(\beta u)} du \quad (\text{III.6})$$

$$\leq \lambda(1 - \gamma) \int_{\beta x}^{\infty} q(u) e^{-\gamma Q(u)} du \quad (\text{III.7})$$

$$\leq \frac{\lambda(1 - \gamma)}{\gamma} e^{-\gamma Q(\beta x)} \quad (\text{III.8})$$

This shows that $C(x)$ is upper bounded by $C(0)$. (Note that for the case of TCP, $\beta = 2$ and $\lambda = 1$, so that $C(0) = 1$; in other cases, $C(0)$ is some finite value.) Now, consider a random variable with density $f(x) = \frac{J(x)}{C(0)}$ and let X_1, X_k, \dots, X_k be k i.i.d realizations of this random variable and let $X_{(1)}, X_{(2)}, \dots, X_{(k)}$ be the order statistic. Then,

$$\overbrace{\int_{x_1 > x} \dots \int_{x_k > a(x_{k-1})} J(x_1) \dots J(x_k) dx_k \dots dx_1}^{k\text{-fold}} = \frac{\{C(x)\}^k}{k!} \times$$

$$Prob(X_{(j)} > a(X_{(j-1)}) \text{ for } j \in (2, \dots, k) | X_j > x, \forall j). \quad (\text{III.9})$$

Hence, if we denote the sum of the first l terms in the RHS of equation (19) as $H_l(x)$, we see that

$$|H(x) - H_l(x)| \leq \bar{H} \sum_{j=l}^{\infty} \frac{C(x)^j}{j!}, \quad (\text{III.10})$$

which proves that $H(x)$ is indeed convergent.

REFERENCES

- [1] V. Jacobson and M. Karels, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM'88, August 1988.
- [2] T. Ott, M. Matthys and J. Kemperman, "The Stationary Behavior of Ideal Congestion Avoidance", [http://web.njit.edu/~ott/Papers/Mathis/ TCP-window.pdf](http://web.njit.edu/~ott/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 Window Size 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. Matthys, 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/~ott/Papers/ECN/ECN.pdf>.