Comprehensive Performance Analysis of a TCP Session Over a Wireless Fading Link With Queueing

Alhussein A. Abouzeid, *Member, IEEE*, Sumit Roy, *Senior Member, IEEE*, and Murat Azizoglu, *Member, IEEE*

***Abstract*—A link model-driven approach toward transmission control protocol (TCP) performance over a wireless link is presented. TCP packet loss behavior is derived from an underlying two-state continuous time Markov model. The approach presented here is (to our knowledge) the first that simultaneously considers 1) variability of the round-trip delay due to buffer queueing; 2) independent and nonindependent (bursty) link errors; 3) TCP packet loss due to both buffer overflow and channel errors; and 4) the two modes of TCP packet loss detection (duplicate acknowledgments and timeouts). The analytical results are validated against simulations using the *ns*-2 simulator for a wide range of parameters; slow and fast fading links; small and large link bandwidth-delay products. For channels with memory, an empirical rule is presented for categorizing the impact of channel dynamics (fading rate) on TCP performance.**

***Index Terms*—Performance analysis, transport control protocol, wireless communications.**

I. Introduction

TRANSMISSION control protocol (TCP) is currently the most widely used transport protocol in packet networks [1] such as the Internet and is largely responsible for end-to-end congestion control. It achieves its congestion control objectives by slowly increasing the rate at which a source releases packets (i.e., the transmission rate) into the network while reacting to any detected packet loss (or explicit congestion notification [2]) by reducing its transmission rate.

In this paper, we provide a comprehensive performance (goodput) analysis of a single TCP connection for a mobile host/client communicating with a server in a backbone (wired) network. Such scenarios are becoming increasingly commonplace due to the advent of wireless access to the Internet; in such cases, TCP goodput can be limited due to losses *both* at the wireless (last) hop, as well as the network backbone due to congestion. However, the literature on TCP modeling to date has typically concentrated on one of the two loss mechanisms, i.e., 1) *channel loss* (errors attributable to the wireless channel) and 2) *congestion loss* (packet drops at queues or buffers) to the exclusion of the other (by assuming it to be negligible). Congestion loss models apply to a bottleneck link in an end-to-end path with multiple traversing TCP flows. A primary contribution of our work is to provide a framework where *both* losses can be treated as is appropriate for wireless Internet access where the channel loss due to (time-varying) multipath fading on the wireless hop can be dominant (or comparable to) congestion losses. A preliminary distinction between *queueing loss* and *channel loss* is possible based on the fact that while packet drop (for Drop Tail policy, as assumed in this work) is inevitably correlated, channel errors can vary from random (independent, isolated packet losses) to bursty (correlated errors).

A useful abstraction for wireless loss scenarios is a bulk TCP transfer over a single link (i.e., queue plus channel) between a source/destination pair that is subject to channel-induced packet loss between TCP sender and receiver (e.g., [3]–[6]). On the other hand, the typical abstraction used for congestion loss is an *equivalent* per flow random loss model that randomly drops packets at the buffer[1] (e.g., [7]–[11]). Note that while both abstract models are superficially the same, the causative loss mechanisms are quite different in nature.

In the pioneering work [12], Lakshman and Madhow provide a complete analytical description of the (now well-known) periodic TCP congestion window evolution over *ideal* (non lossy) channels with finite buffer size, which provides a point of departure for part of our work. They also consider a packet-level random loss model where every transmitted packet may be lost with a probability q independent of all other packets. Fixed point approximations are used to arrive at general rules for buffer dimensioning—we remark that use of such fixed point approximations do not provide accurate estimates for TCP throughput in the presence of high correlation between the individual flow's congestion window size and the round-trip time (RTT) [7]. Timeout effects are also neglected in the analysis of the random packet loss case.

Our work is philosophically closer to the broad renewal-reward based approach espoused in [4]–[6]. We note that [3] uses an independent and identically distributed (i.i.d.) packet-level loss model while [4]–[6] use *correlated* packet-level loss models based on a *discrete time* two-state Markov chain to model the wireless channel errors [13]. Our analysis has the following unique contributions.

A. A Channel Driven Model for Packet Loss

Previous wireless channel models for packet loss *implicitly* assume that the TCP sources are *continuously transmitting* and ignore the fundamental bursty nature of TCP transmissions.

[1]This represents losses over an end-to-end path with multiple links shared by a large number of TCP sessions.

Manuscript received March 28, 2001; revised November 22, 2001 and April 19, 2002; accepted April 26, 2002. The editor coordinating the review of this paper and approving it for publication is M. Zorzi.

A. A. Abouzeid is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: abouzeid@ecse.rpi.edu).

S. Roy is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195-2500 USA (e-mail: roy@ee.washington.edu).

M. Azizoglu is with Sycamore Networks, Inc., Chelmsford, MA 01824-0986 USA (e-mail: murat.azizoglu@sycamorenet.com).

Digital Object Identifier 10.1109/TWC.2003.808976

This is tied to the alternating two-state channel model currently implemented in *ns*-2 where the *state evolution is clocked by packet transmissions; thus, the link state does not advance in time when TCP sender is idle*. This contradicts the actual physical behavior since channel state evolution is not predicated by packet transmissions and renders purely *ns*-2 based simulation analysis suspect without independent corroboration. Previous TCP over wireless channel models such as in [6] impose a distribution on the *number* of consecutive packet drops in a bad state *regardless of the time duration between the transmission of these packets*. This assumption is only valid for continuous packet transmission; since TCP is a bursty protocol, the intermittent idle durations during the lifetime of a connection may be considerable (unless the window size is always equal to the bandwidth-delay product of the link, which is a very limiting scenario) thereby undermining this key assumption. In this paper, we present a stochastic channel-driven model of TCP over wireless links that does not make any prior assumptions on the packet-level loss statistics; these are instead derived from the underlying physical channel behavior. The derivation accounts for the effects of idle periods which are an essential characteristic of TCPs bursty nature.

B. Modeling the Queue Behavior

We underscore that previous wireless loss models neglect the effect of the queue behavior. Specifically, *all* previous *correlated* loss models consider the case of very high wireless channel loss rates and hence neglect queue losses. The queue behavior can have significant effect on TCP goodput, especially in situations in which a transmit buffer is dedicated solely for the use of the TCP session (practical situations are discussed in Section II). Such situations can be characterized by the following properties.

1) High correlation between window size and RTT. The typical correlation coefficient in such cases can be as high as 0.97 [7]. This degrades the accuracy of models that assume a fixed RTT independent of the window size (see also the discussion of this issue in [7]).
2) Possible additional packet loss due to buffer (queue) overflow. When channel losses take place at a sufficiently low rate, packet queueing may take place that may exceed the capacity of the queue.

Our model accounts for both of the above two characteristics; we allow the RTT to be a function of the window size (rather than fixed) and we model both types of packet loss; queue loss (buffer overflows) and channel (wireless) losses.

C. A Unified Model for Various System Parameters

Our model is applicable to a wider range of network parameters than previous related work. Specifically, in [3], the effects of possible queueing losses were neglected, the binary exponential backoff algorithm is approximated to only two consecutive timeouts, a local-area network (LAN) environment is assumed (hence, the round-trip propagation delay is set to zero for an acknowledgment (ACK)), each ACK during the congestion avoidance phase is assumed to cause the window size to be incremented by one with a certain probability and the packet transmission time at the bottleneck link is assumed to be exponentially distributed. Ref. [4] assumes a buffer with infinite capacity (no packet loss due to buffer overflow), [5] considers links with "zero" delay (instantaneous acknowledgments) and "no queueing," and [6] neglects the variability in the RTT due to the queue dynamics in addition to neglecting packet losses due to queue overflow. Our model thus provides a unified approach for all these situations. Finally, our previous work in [14] models the RTT variability in the queue size, as well as packet losses due to queue overflow but assumes the loss statistics to take place at the packet level, which is a drawback in all the previous models (simulation and analytically based) as mentioned earlier.

Models for congestion loss bear striking resemblance to those for channel loss described above; these typically neglect link impairments and seek to describe queue loss due to many shared TCP flows at the network queues as a "random" (i.i.d.) packet drop phenomena with an associated fixed probability. Mathis *et al.* [8] model such packet losses initially on the assumption that the RTT is constant and extend it to include the (variable) queueing time by using the average measured RTT. Thus, their model requires that both the average RTT and the packet loss probability must be known. Other significant extensions to [8] also use the same assumptions of constant queueing delay (e.g [7], [9]–[11]). Congestion loss models (whether correlated or i.i.d) *cannot* be used for modeling the TCP dynamics over a wireless link since they assume: 1) a known loss process at the packet layer (i.e., they do not attempt to derive the packet loss behavior from the underlying channel statistics) and 2) the only source of packet loss is congestion (i.e., they do not attempt to differentiate between the source of packet loss)—and hence the analytical results from such models tend to overestimate the throughput as compared with simulations in which wireless as well as congestion loss exist.

It is worth mentioning that in [15], the authors present an analysis of the case where link layer retransmissions are used to hide link-level errors from TCP. While this is not within the scope of our work, we highlight the differences between the approach in [15] and that adopted in this paper: 1) [15] assumes that "the arrival process of TCP packets at the wireless link is modeled by a stationary Bernoulli process" and "this assumption is made primarily because an exact analysis accounting for the dynamics of TCP and of the wireless channel is intractable" ([15, pp. 608, Sec. II-C]). On the other hand, our model yields effective (the computations take two orders of magnitude less time than the simulations), tractable and accurate results that do not make such assumptions on the bursty window process of TCP and 2) Unlike the referenced paper, which uses a custom-made simulator for model validation, our analysis results are validated against the publicly-available network simulator *ns*-2 [16].

The remainder of this paper is organized as follows. The model for TCP and a summary of the previous results for modeling TCPs congestion avoidance algorithms over ideal channels are included in Section II. Section III presents the loss model used in this paper for the two cases of loss considered (i.i.d and correlated). Section IV presents a unified analysis for the case of i.i.d. as well as correlated loss. Section V describes the simulation set-up and presents representative validation

results. Section VI contains final remarks and concludes by outlining possible suggestions for future work.

II. A Model for TCP Congestion Avoidance

This section introduces our abstract system model, the model for TCP congestion avoidance, previous well-known analytic characterization of TCPs behavior for ideal channels and our model for TCPs bursty nature. The section is essential for following the notation in the rest of the paper since we introduce most of the notation and definitions of various terms.

A. System Model

Consider a mobile host (client) connected to a host (server) in the wired backbone via a last-hop wireless link. We assume that the wireless link constitutes the bottleneck on that path and that the remainder of the connection through the network can be simply modeled as a constant delay. The round trip time τ for a TCP packet is the time elapsing between its release from the source and the reception of the corresponding acknowledgment (ACK) for a successful transmission, (for simplicity, it is assumed that ACKs are not lost, as is reasonable for symmetric links with cumulative TCP ACKs) excluding processing and queueing times at the wireless bottleneck link. The wireless link has a raw capacity of μ packets/s and buffer size B packets devoted to the connection of interest. This could arise, for example, in a time-division multiple-access (TDMA) system with static bandwidth assignments or in the case of a congested last-hop connection from a (mobile) user to an ISP ([17]).

Define $T = \tau + 1/\mu$ to be the time (in seconds) between the start of a packet transmission and reception of a corresponding ACK, *excluding* any queueing delays in the buffer. Then, μT is the bandwidth-delay product and the ratio $\beta = B/(\mu T)$ is the normalized buffer size by the bandwidth-delay product. Let w_p denote the maximum number of packets possibly in transit between the source and destination (including the packets in the link buffer); thus, $w_p = B + \mu T = \mu T(\beta + 1)$. This basic model or some variant has been used in [3]–[5], [7], [8], and [12].

B. TCP Operation in Ideal Channels

The window-based congestion avoidance mechanism in TCP/IP [17] acts as a self-clocking regulator based on receiver feedback (or lack thereof). In this paper, we assume the reader is familiar with the window adaptation mechanism of TCP-Reno: the two modes of window increase; slow start and congestion avoidance, the two modes of packet loss detection; reception of multiple (typically four) consecutive ACKs with the same next expected packet number (abbreviated as triple duplicate or TD) or timer-expiry (time outs, abbreviated as TO) and the binary exponential backoff algorithm. We assume the receiver sends an ACK for each packet (i.e., no delayed ACKs).

It is well-known that, for ideal channels (i.e., no random packet loss), TCP exhibits a periodic evolution [12]. Let $t' = 0$ denote the time of establishment of the TCP-Reno session under consideration. $W(t')$ and $W_{th}(t')$ represent the congestion window size and the slow start phase threshold at time t', and $\Delta(t')$ is the current timeout value. The main observations for TCP Reno window evolution can be summarized as follows.

TABLE I
Definitions of Various Parameters Used for Computing the Reward and Transition Probabilities

Parameter	$H = A$	$H = B$
$t(w_I, w_F, H)$	$T(w_F - w_I)$	$\frac{w_F^2 - w_I^2}{2\mu}$
$t_n(w_I, H)$	$T(\sqrt{w_I^2 + 2n} - w_I)$	$\frac{n}{\mu}$
$n(w_I, w_F, H)$	$\frac{w_F^2 - w_I^2}{2}$	$\frac{w_F^2 - w_I^2}{2}$

TABLE II
Duration and Number of Packets Successfully Transmitted During the Typical Cycle

Parameter	$\beta < 1$	$\beta > 1$
t_A	$t(w_p/2, \mu T, A)$	0
t_B	$t(\mu T, w_p, B)$	$t(w_p/2, w_p, B)$
N_A	$n(w_p/2, \mu T)$	0
N_B	$n(\mu T, w_p)$	$n(w_p/2, w_p)$

1) Except for an initial slow-start phase at the beginning of the session, the congestion window size shows a periodic evolution. Each cycle starts with window size $W(t') = w_p/2$ and continues in congestion avoidance phase until a buffer overflow at w_p, whereupon the window is halved to $w_p/2$.
2) Assuming fast recovery and retransmit options implemented [18], no delayed ACKs, and sufficiently large window size, packet(s) lost due to a buffer overflow is (are) detected via TD. Hence, no TOs take place, and the TCP Reno session remains in congestion avoidance during the lifetime of the session.
3) In congestion avoidance, the window increase is either sublinear ($O(\sqrt{t})$) if $w_p/2 > \mu T$, or a combination of linear followed by a sublinear increase if $w_p/2 < \mu T$. The linear growth is called **congestion avoidance phase A** while the sublinear phase is called **congestion avoidance phase B**.

Let $H = A$ (B) denote the congestion avoidance phase A (B) and w_I and w_F (nonnegative integers with $w_F > w_I$) be the initial and final window sizes during a congestion avoidance phase; $t(w_I, w_F, H)$ the time taken by congestion window size to increase from w_I to w_F during the congestion avoidance phase$H$ assuming no packet losses. The n^{th} packet sent during this interval $n(w_1, w_2, H)$ is associated with $t_n(w_1, H)$, the time of transmission; e.g., $t_0(w_I, H) = 0$,$t_1(w_I, H) = 1/\mu \ldots$, etc. Using the derivation in [12], these parameters are easily computed and are summarized in Table I. Let t_A and t_B denote the duration of the congestion avoidance phases A and B, and let N_A and N_B denote the corresponding number of packets successfully transmitted during these two phases. Let $t_p = t_A + t_B$ denote the duration of a cycle and $n_p = N_A + N_B$ the number of packets successfully transmitted during the cycle. Then, t_A, t_B,N_A, N_B, and n_p can be expressed in terms of the parameters in Table I for the two cases of β ($\beta < 1$ and $\beta > 1$) as shown in Table II. The parameters in Tables I and II will be used in the analysis later in the paper.

Note that in all the above expressions, we only consider time instants t' where $W(t')$ is discrete. This allows us to use a dis-

crete analysis of the window size rather than a fluid approximation, which is well-known to render more accurate results (e.g., see [10] for a quantitative discussion of the effect of fluid based approximations in TCP modeling). Since TCP maintains integer valued window size and number of packets, all the expressions in Tables I and II for these parameters that include fractions are replaced by the appropriate integer value $\lfloor . \rfloor$ for number of packets and $\lceil . \rceil$ for window size.

C. TCP Burst Model

For clarity of presentation, we make the following definitions.

Def. 1—Window Round: A TCP window round (or round, simply) for a window of size w_1 is the time elapsed between the congestion window successive increments from w_1 to w_1+1. If μ is the packet transmission time, then the duration of the round during the congestion avoidance phase A is equal to $\tau + 1/\mu$ and congestion avoidance phase B is equal to $(w_1)/\mu$.

Def. 2—Busy Period of a Round: A busy period of a round w_1 is the time duration, from the beginning of the round, during which TCP sender is busy transmitting packets; equivalently the duration when all packets during this round are transmitted. The duration of this period is equal to w_1/μ.

Def. 3—Idle Period of a Round: In a window round w_1, if $w_1 < \mu T$ (i.e., congestion avoidance phase A), then an idle period of duration $T - w_1/\mu$ terminates the round.

Thus, our burst model for packet transmissions within each window round during the congestion avoidance can be fully described by the above definitions as follows. TCP transmits packet in bursts, the length (expressed in number of packets) of each burst of packets is equal to the window size. Each round for which $W(t') < \mu T$ (congestion avoidance phase A) consists of a busy period followed by an idle period, the duration of each is defined above. As the window size increases (assuming no packet losses), the duration of the busy period within each round increases (and the duration of the idle period decreases) until the window reaches μT, at which point the whole duration of the round is consumed by packet transmissions. When the window size exceeds μT, the duration of the round increases in steps of $1/\mu$ for each window increment in excess of μT.

III. Channel Model

The wireless channel is modeled by a continuous time, two-state alternating process $\{S(t')|t' \geq 0\}$ taking values $k = 0$ (good) or 1 (bad) with properties that:

1) when the channel leaves one state, it will enter the other state with probability 1 (i.e., an alternating process);
2) the durations of time in the good state, denoted$\{X_i, i = 1, 2, \ldots\}$, are i.i.d. with known exponential cumulative distribution function $F_X(x)$ and mean $E[X_i] = 1/(\lambda_0)$;
3) the durations of time in the bad state, denoted$\{Y_i, i = 1, 2, \ldots\}$, are i.i.d. with known exponential cumulative distribution function $F_Y(y)$, mean $E[Y_i] = 1/(\lambda_1)$, and independent of the $\{X_i\}$;
4) in each of the states, the packet loss mechanism is that of a discrete memoryless (packet) channel, with respective loss probabilities p_0 and p_1. For simplicity, in this work we assume that $p_0 = 0$, $p_1 = 1$, i.e., a packet is successfully transmitted only if the channel at the receiver remains in the good state for the duration of the packet reception, otherwise, the packet is lost. Note that this approximation can be further refined by considering the specific link layer detector that typically averages/integrates the received signal over a packet duration prior to declaring reception/loss.

Such a channel state model has been justified by earlier work notably [5] and is widely accepted for transmissions from a base station to a mobile surrounded by scatterers. The model for successful transmission in the good state and unsuccessful transmission in the bad state is based on the fact that the performance of most link-layer error correction or detection codes is good if the signal power is above a certain threshold, while it deteriorates rapidly once the signal power falls below the threshold. Given this power threshold and the Doppler frequency, the duration of the good and bad states are modeled as some (usually exponential) random variables whose means can be estimated using the fading model. This results in the above two-state channel model that is widely used for performance predictions. For instance, for fading with 10 Hz Doppler frequency,[2] suppose that the bad state corresponds to a signal power less than 10 dB below the nominal signal power. This can be used to deduce that the mean duration of the good state is roughly 100 ms and that of the bad state is roughly 10 ms.

Note that a difficulty with the above model is that it cannot be directly translated to packet loss statistics. For example, the average good duration $E[X_i]$ does *not* correspond to the average successful packet transmission time (similarly for the average bad duration)— a visit to the bad state may not incur any packet losses if the TCP session does not attempt to transmit packets during this duration. *We underscore this fundamental point since it appears to have been missed in some previous work—the channel state evolution is independent of TCP dynamics but* ***not*** *vice versa*. A one-to-one correspondence only holds if TCP is continually transmitting packets (i.e., the case of a fully utilized link).

For future reference, we introduce the (long-term) fraction of time the channel spends in the bad state f given by

$$f = \frac{E[Y]}{E[X] + E[Y]}. \tag{1}$$

IV. Analysis

We start the analysis by identifying the key characteristics of TCP behavior as they relate to an underlying lossy channel motivated by *ns*-2 simulation traces. Subsequently, we describe a semi-Markov process that closely models this TCP behavior within a number of well-motivated approximations. A Markov renewal-reward approach is then applied toward deriving the steady-state window distribution and goodput.

[2]The rate of channel variations is characterized by the maximum Doppler frequency.

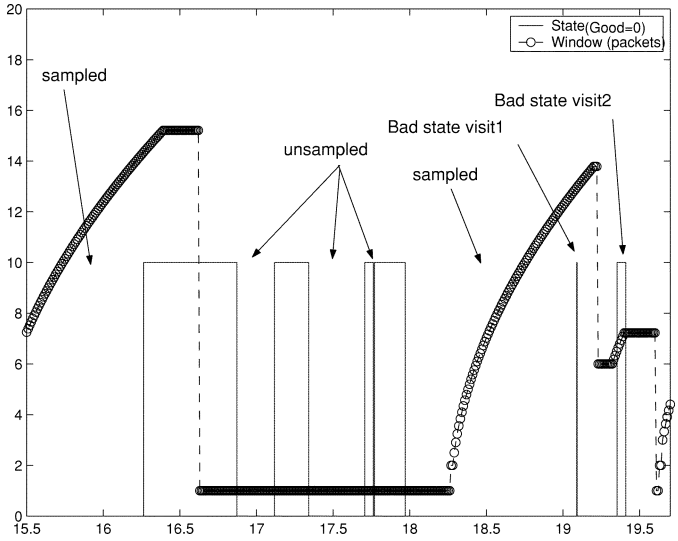


Fig. 1. A simulation trace for the window size and channel state against time using the channel-driven loss model implementation in *ns*-2. ($\mu = 100$, $\beta = 8.0$, $\tau = 0.01\,\mathrm{s}$, $B = 16$ packets, $w_p = 18$).

A. Key Characteristics of TCP Over a Channel-Driven Loss Model

Correlated loss channels are characterized by a *finite* (as opposed to infinitesimal) bad state duration; consequently, packet losses can occur in bursts (as well as in isolation). In this case, packet loss detection occurs both via TDs and TOs. For sufficiently high loss rates and/or small congestion windows, TOs may occur even for i.i.d. loss. While these observations hold in general for any wireless (correlated) loss model, we identify the following *additional* key characteristics for a channel-driven loss model.

1) Channel visits to the good state during which TCP transmits packets (i.e., contribute to TCPs goodput) are "sampled" good states.
2) Channel visits to the good state during which TCP does not attempt to transmit packets are "unsampled" (or skipped) good states.
3) A bad state visit may result in a TO, TD, or no packet loss (NL).
4) In addition to channel induced losses, buffer overflow may also take place if the window reaches w_p.

Most of the above characteristics are evident in the *ns*-2 simulation trace Fig. 1. Initially, the channel is in a (sampled) good state and TCP is successfully transmitting packets. A sufficiently long bad state duration follows, causing a loss of enough packets to trigger a TO and, hence, a retransmission of these packets. However, these packets are also lost since the channel remains in the bad state. Notice that TCP has to wait for the next TO before attempting to retransmit again. During this time, a channel visit to the good state may not be detected resulting in an "unsampled" (or skipped) good state. Due to TCPs binary exponential backoff algorithm (see Section II for a summary), the timer expiry value upon each consecutive TO doubles (up to a ceiling value), hence allowing for longer duration good states to be skipped. In the simulation trace in Fig. 1, three good states are unsampled. The next good state has sufficient duration to be sampled, and TCP enters again into congestion avoidance with an initial window size equal to one. The bad state visit labeled visit 1 is of a very short duration and results in a short burst of packet losses which is detected by a TD. The subsequent bad visit (visit 2) is long enough to cause a TO.

Note that due to the bursty nature of TCP source, a visit to the bad state does not necessarily result in a packet loss, since it may occur during the idle period of the round (for simplicity, this assumption was made in our earlier work [14] resulting in lower model accuracy, especially for the case where $\beta < 1$). This does not take place in the simulation trace since all bad states are of relatively long duration. Finally, the simulation trace in Fig. 1 shows only channel induced losses. However, queue losses may also take place if the window size reaches w_p, such queue losses do not occur in the reported trace since the window does not reach w_p due to frequent bad state visits.

B. Modeling Approach

TCPs dynamics exhibit a large amount of memory. For example, TCPs algorithm for setting the timer-expiry period $\Delta(t')$ retains all RTT samples since the start of the TCP session. The memory is also evident in the delay in TCPs TD algorithm, which depends in a nontrivial way on the specific location of the packet(s) lost within a window round (e.g., see [18]).

In order to track the joint "TCP/channel" evolution precisely by a Markov random process (i.e., a process in which the future evolution is completely determined by the current state), one needs to define a process $C(t') = (S(t'), W(t'), W_{th}(t'), \Delta(t'), \Gamma(t'))$, where $\Gamma(t')$ denotes the "age" of the round (i.e., the time elapsed since the beginning of the current round). Clearly, this produces such a large state space as to render the approach impractical.

Our analysis is based on applying Markov renewal-reward theory based on the following approximations to the actual TCP/channel evolution.

1) The delay between a packet loss event and its detection—of the order of one half of a RTT for a symmetric link—is neglected. This results in a slight overestimation of the goodput; in practice, the number of packets successfully transmitted in the interval between packet loss and its detection may be also lost and hence may not contribute to the goodput.
2) We assume only *one bad state visit per round* which suffices when the good and/or bad state durations are sufficiently long. For multiple transitions between good and bad states during the same round (as when the state durations are short), our model provides a lower bound since such multiple bad state visits are reflected in our model as visits to different rounds. Further, our model provides a closer match to Reno than New Reno version, since New Reno treats multiple losses within the same window round as only one loss. However, our validation experiments considered the case where this assumption is not valid, and we found our analytical estimates to still be reasonably accurate.
3) We assume the timer expiry value at the end of a sampled good state, i.e., $\Delta(w_2)$ is completely determined by the window size w_2. In reality, $\Delta(t')$ is based on a smoothed

estimated of the RTT using an exponentially weighted moving average—thus effectively having memory from the start. Our assumption is motivated by well-known simulations that show that for fine-grained timers, $\Delta(t')$ approaches the true RTT for slowly varying RTTs (e.g., [19]), i.e., $\Delta(t')$ is close to the current RTT for sufficiently long sampled good states. Thus

$$\Delta(w_2) = \frac{1}{\mu}\max(w_2, \mu T). \tag{2}$$

Reducing $\Delta(t')$ to the most recent window size value preserves the embedded MC description complexity to a minimum.

4) Instead of keeping track of the exact instant within a window round (i.e., age) at which a sampled good state ends (by visiting a bad state), we assume an origin of the bad state duration uniformly distributed over the duration of the round. Specifically, let $\Gamma(w_2)$ denote the origin of the bad state duration, conditioned on a visit to the bad state during the round w_2. Then

$$F_\Gamma(\gamma|w_2) \sim \mathrm{U}\left(0, \max\left(T, \frac{w_2}{\mu}\right)\right] \tag{3}$$

where $U(a, b)$ denotes the uniform distribution.

5) Following a TO, TCP-Reno reverts to slow start, where the window is set to one and the slow-start threshold is set to half the window size at which TO is detected. If further TOs occur, the threshold window will be successively reduced toward the limit 1. We assume that upon a single TO event, the window is set to one and that TCP-Reno *immediately* enters the congestion avoidance phase. This is equivalent to approximating the slow start phase in the case of a single TO by a linear window increase instead of an exponential increase; the impact is a predicted goodput slightly less than the true.

The above approximations render an idealized TCP process that is a close replica of the actual one. This new process is effectively modeled as a semi-Markov process for which two embedded Markov chains (MCs) can be identified and analyzed with a minimal number of states, as presented in the following subsection.

C. The Embedded MCs

Let W_n denote the window size at the instant a good state X_n is first sampled. Then the remaining time in the good state V_n is also exponentially distributed with parameter λ_0, by the memoryless property of the exponential distribution. Conditioned on the value of W_n, the window size evolution until the sampled good state terminates (due to a visit to the bad state) is completely independent of the past. Let the window size at the end of the sampled good state be W'_n. Then, similarly, conditioned on the value of W'_n, the future evolution of the congestion window is independent of the past. The duration of time between successive sampled good states may include multiple unsampled good states in addition to bad states. The latter can be lumped into a single "effective bad duration," denoted U_n for analytical purposes. Since X_i and Y_i are independent, V_n and U_n are also independent. Thus, the sequence $\{W_n\}_{n=1}^{\infty}$ of the process defined just after a good state is first sampled form an embedded MC (of the semi-Markov process). Similarly, the sequence $\{W'_n\}_{n=1}^{\infty}$ of the process defined just before a sampled good state terminates (or, equivalently, an effective bad duration starts) form an embedded MC. The former is called the "good" MC and the latter is called the "bad" MC.

Notice that each of the previously mentioned approximations to the actual TCP process is necessary for above embedded processes to be Markov, and the inclusion of any of the approximated behaviors is only possible at the cost of increasing state space (i.e., the dimensionality) of the resulting embedded MCs—this proves to be unnecessary as shown by comparisons against *ns*-2 simulations of the actual TCP process reported in Section V.

Three quantities associated with the embedded MCs are of interest.

1) $E[N|W_n = w_1]$, the expected number of packets successfully transmitted (the reward) during a sampled good state given that it started at a window w_1. Packet transmissions/retransmissions during the effective bad states have zero contribution toward TCPs goodput and thus the reward during an effective bad duration is zero.[3]
2) $P[W_{n+1} = w_3|W_n = w_1]$ (denoted P_{w_1,w_3}); the probability that the next sampled good state starts at a window w_3 given that the previous sampled good state started at a window w_1, or equivalently, the transition probabilities of the "good" MC. Also of interest is $P[W'_{n+1}|W'_n]$, the transition probabilities of the bad MC.
3) $E[U]$, the expected duration of time between the end of a sampled good state and the beginning of the following one (i.e., the average effective bad duration).

We define *the typical cycle* as the time duration between two consecutive buffer overflows between which no bad state visits take place. An *atypical cycle* is the duration upon first sampling a good state in which the window increases from a starting value of w_1 until w_p. Thus, between any two consecutive sampled good states, TCP-Reno starts by entering the atypical cycle; depending on the duration of the good state, one or more typical cycles (buffer overflow events) follow until the subsequent visit to the bad state.

The ranges of β ($\beta < 1$ and $\beta > 1$) are considered separately for the purpose of computing the various parameters of interest due to the differences in the typical cycle (for $\beta > 1$, the typical cycle consists of a congestion avoidance B phase only). Also, for each range of β, the cases of $w_1 < \mu T$ and $w_1 > \mu T$ are considered separately due to the differences in the atypical cycle (for $w_1 > \mu T$, the atypical cycle consists of a congestion avoidance B phase only).

D. Calculating Conditional Reward $E[N|W_n]$

The derivation for $\beta < 1$ is presented here and that for $\beta > 1$ follows in a similar fashion:

$$E[N|W_n = w_1] = \sum_{m=0}^{\infty} Pr[N > m|W_n = w_1]. \tag{4}$$

[3]While the focus in this paper is on calculating the goodput, it is also possible to calculate the throughput by computing the (small) reward during the effective bad duration.

Consider the case $w_1 < \mu T$. During an atypical cycle, $0 \leq m < n(w_1, \mu T, A)$ and $0 \leq m < n(\mu T, w_p, B)$. For the jth (where $0 \leq j < \infty$) typical cycle, the number of packets sent satisfy $0 \leq m < N_A$ and $0 \leq m < N_B$. Hence, in terms of the complementary distribution function of X, i.e., $\bar{F}(a) = Pr[X > a]$

$$\begin{aligned} s_1(w_1) = & \sum_{m=0}^{n(w_1,\mu T,A)-1} \bar{F}[t_{m+1}(w_1, A)] \\ & + \sum_{m=0}^{n(\mu T,w_p,B)-1} \bar{F}[t(w_1, \mu T, A) + t_{m+1}(\mu T, B)] \\ & + \sum_{j=0}^{\infty} \sum_{m=0}^{N_A-1} \bar{F}\Big[t(w_1, \mu T, A) + t_B + jt_p + t_{m+1}\left(\frac{w_p}{2}, A\right)\Big] \\ & + \sum_{j=0}^{\infty} \sum_{m=0}^{N_B-1} \bar{F}[t(w_1, \mu T, A) + (j+1)t_p + t_{m+1}(\mu T, B)] \end{aligned} \tag{5}$$

where we introduce $s_1(w_1)$ for notational convenience. The first term in (5) represents the probabilities that the good duration exceeds the time required to send $m \leq n(w_1, \mu T, A)$ packets during the congestion avoidance A phase of the atypical cycle. The second term represents the probabilities that the good duration exceeds the time required to send $n(w_1, \mu T, A) < m \leq n(w_1, \mu T, A) + n(\mu T, w_p, B)$. The remaining two terms are computed in a similar fashion.

The result for $w_1 > \mu T$, denoted $s_2(w_1)$ is similar to (5), with the difference that the atypical cycle consists only of congestion avoidance B phase. Combining the two cases

$$E[N|W_n = w_1] = \begin{cases} s_1(w_1) & w_1 < \mu T \\ s_2(w_1) & w_1 > \mu T \end{cases}. \tag{6}$$

E. The Transition Probability $\Pr[W_n'|W_n]$

Computation of P_{w_1,w_2} follows the same steps as above for the reward, i.e., we also condition on a sampled good state at w_1 and depending on X the visit to the bad state can take place while the window size is w_2 during the atypical cycle (only if $w_2 > w_1$), or during any of the following $0 \leq j \leq \infty$ typical cycles. This can be expressed in terms of $\bar{F}(x)$, where we introduce $\tilde{F}(a(w_2)) = \bar{F}(a(w_2), a(w_2 + 1)) = \bar{F}(a(w_2)) - \bar{F}(a(w_2 + 1))$. For example, for the case $\beta < 1$, $w_1 < w_p/2$ and $w_p/2 < w_2 < w_1$, $P_{w_1,w_2} = \tilde{F}(t(w_1, w_2, A))$.

F. Outcome of a Visit to a Bad State

As mentioned earlier, a sampled good state terminates upon a visit to a bad state with three possible outcomes—no packet loss (NL), packet loss(es) that trigger a TD, or packet loss(es) that trigger at least one TO. This subsection evaluates the probability of each of these events, conditioned on the sampled good state terminating at a window size w_2.

The NL case occurs only if the subsequent bad state is sufficiently short and takes place during the idle time of the w_2 round. This implies $w_2 < \mu T$, otherwise the round will be fully occupied by packet transmissions. It follows that $\Gamma(w_2)$ is uniformly distributed in $[0, T]$ [see (3)]. Let $P_{\mathrm{NL}}(w_2)$ denote the conditional loss probability. Using Fig. 2(a)

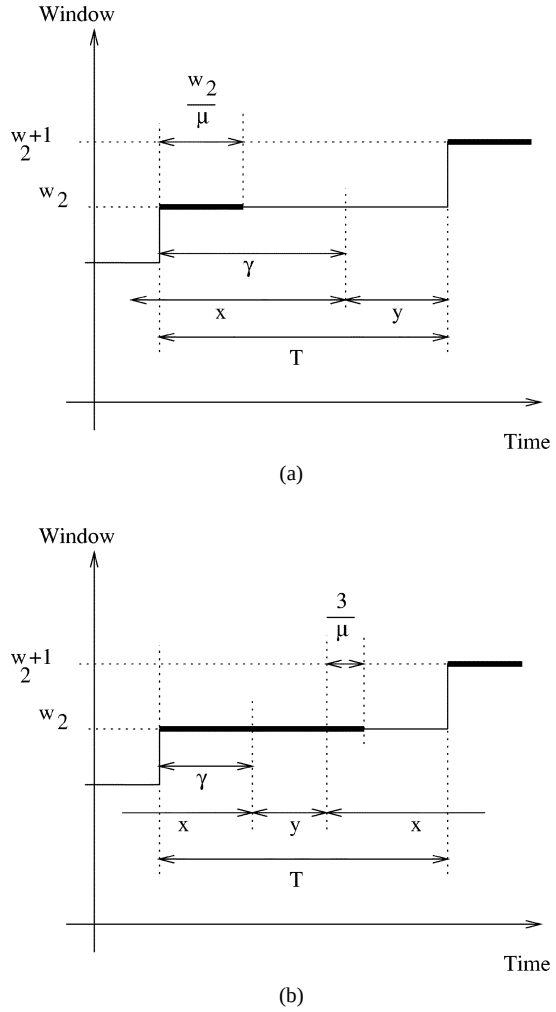


Fig. 2. Sketch of two cases (for $w_2 < \mu T$) where the visit to the bad state (a) does not result in any packet loss and (b) results in packet losses that are detected by TD.

$$P_{\mathrm{NL}}(w_2) = \begin{cases} \int_{\gamma=\frac{w_2}{\mu}}^{T} \overline{F}_{\Gamma}(\gamma) F_Y(T-\gamma) d\gamma, & w_2 < \mu T \\ 0, & w_2 > \mu T. \end{cases} \tag{7}$$

For the TD case, first note that if $w_2 < 4$, any packet loss will result in having less than three packets in transit and, hence, cannot possibly generate a TD. Next, consider $3 < w_2 < \mu T$—then a TD will take place only if less than $w_2 - 3$ packets are lost from the round w_2 and at least three packets are transmitted successfully thereafter. This yields (referring to Fig. 2(b) for the case of $3 < w_2 < \mu T$)

$$P_{\mathrm{TD}}(w_2) = \begin{cases} 0; & w_2 \leq 3 \\ \left(\int_{\gamma=0}^{\frac{w_2-3}{\mu}} F_Y\left[\frac{w_2-3}{\mu} - \gamma\right] dF_{\Gamma}(\gamma) \right) \bar{F}_X\left[\frac{3}{\mu}\right]; & 3 < w_2 \leq \mu T \\ F_Y\left[\frac{w_2-3}{\mu}\right] \bar{F}_X\left[\frac{3}{\mu}\right]; & w_2 > \mu T \end{cases} \tag{8}$$

Finally

$$P_{\rm TO}(w_2) = 1 - P_{\rm NL}(w_2) - P_{\rm TD}(w_2) \tag{9}$$

G. Trans. Prob. $\Pr[W_{n+1}|W_n]$ and $\Pr[W'_{n+1}|W'_n]$

The calculation of P_{w_2,w_3} follows directly from the above derivation:

$$P_{w_2,w_3} = \begin{cases} P_{\rm NL}(w_2), & \text{if } w_3 = w_2 \\ P_{\rm TD}(w_2), & \text{if } w_3 = \frac{w_2}{2} \\ P_{\rm TO}(w_2), & \text{if } w_3 = 1. \end{cases} \tag{10}$$

Since $\Pr[W_{n+1}|W_n, W'_n] = \Pr[W_{n+1}|W'_n]$, we have

$$\Pr[W_{n+1} = w_3 | W_n = w_1] = P_{w_1,w_3} = \sum_{w_2=1}^{w_p} P_{w_1,w_2} P_{w_2,w_3} \tag{11}$$

$Pr[W'_{n+1}|W'_n]$ can be computed in a similar fashion.

H. Steady-State Window Size Distributions $\pi_g(w)$ and $\pi_b(w)$

The steady-state distribution $\pi_g(w)$ of the "good" embedded MC $\{W_n\}_{n=1}^{\infty}$ (describing the window size at the beginning of a sampled good state) can be easily numerically evaluated using MATLAB routines for solving the eigenvalue[4] problem of the matrix of transitions $\Pr[W_{n+1}|W_n]$ (derived above) for different values of λ_0, λ_1, μ, τ, and B. Similarly, the steady-state distribution $\pi_b(w)$ of the "bad" embedded MC $\{W'_n\}_{n=1}^{\infty}$ (describing the window size at the end of a sampled good state- or equivalently the start of an effective bad state), is numerically evaluated.

I. The Average Effective Bad Duration

Consider a sampled good state[5] terminating at a window size $W'_n = w_2$. Let $\Delta(w_2)$ denote the timer expiry at the end of the sampled good state. The three possible outcomes of the bad state visit and the associated probability of each were derived in the previous section. The effective bad duration will be equal to the current bad duration in the first two cases (i.e., NL or TD) and may exceed it in the third case.

The average effective bad duration $E[U]$ is related to the conditional expected bad durations $E[U|\text{NL}, w_2]$, $E[U|\text{TD}, w_2]$ and $E[U|\text{TO}, w_2]$ via

$$\begin{aligned} E[U|w_2] =& E[U|\text{NL}, w_2]P_{\rm NL}(w_2) + E[U|\text{TD}, w_2]P_{\rm TD}(w_2) \\ &+ E[U|\text{TO}, w_2]P_{\rm TO}(w_2) \end{aligned} \tag{12}$$

The calculation of the first two terms is straightforward since we observe that $E[U|\text{NL}, w_2] = E[Y|\text{NL}, w_2]$ and $E[U|\text{TD}, w_2] = E[Y|\text{TD}, w_2]$. We now proceed to derive $E[U|\text{TO}, w_2]$. First, observe that the delay between the visit to the bad state and the timer expiry is equal to $\Delta(w_2)$. Upon the first TO, TCP will retransmit the packet, set its timer-expiry period to $2\Delta(w_2)$ and wait for ACKs. If the retransmission is successful, TCP will resume its transmission, else, another TO will take place followed by a retransmission and the timer expiry will now be set to $4\Delta(w_2)$. This doubling of the TO expiry period continues up to $64\Delta(w_2)$ after which it remains constant. This process of timer expiry and unsuccessful retransmission continues until TCP successfully transmits a packet and the effective bad state, hence, terminates (see Fig. 3).

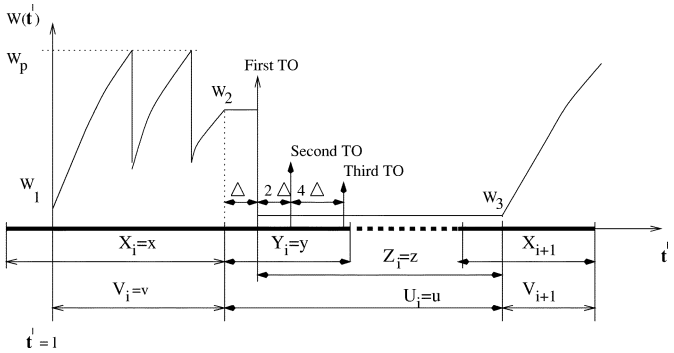


Fig. 3. Sketch of TCP behavior for a case with multiple consecutive TOs.

Let Z denote the time duration spent in a sequence of one or more TOs. Then, the exact number of TOs can be calculated from a transient analysis of the two state MC by sampling the MC state at a rate equal to $\Delta(w_2), 2\Delta(w_2), \ldots, 64\Delta(w_2), \ldots$

$$\begin{aligned} P_\delta =& \Pr[S(l+\delta) = 0 | S(l) = 1] = \Pr[S(\delta) = 0 | S(0) = 1] \\ =& (1-f)\left(1 - e^{-(\lambda_0+\lambda_1)\delta}\right) \end{aligned} \tag{13}$$

where we used the memoryless property in (13) together with a first order analysis of the transient behavior of the channel two-state MC [21]. We use Δ instead of$\Delta(w_2)$ in the following derivations for notational convenience. From (13)

$$\Pr[Z = \Delta] = \Pr[S(\Delta) = 0 | S(0) = 1] = P_\Delta \tag{14}$$

$$\begin{aligned} \Pr[Z = 3\Delta] =& \Pr[S(2\Delta) = 0 | S(\Delta) = 1, S(0) = 1] \\ =& P_\Delta(1 - P_\Delta). \end{aligned} \tag{15}$$

where we used the Markov property in (15). In general, the p.d.f. of Z can be computed just as in (15) and the resulting expressions are included in [22]. Consequently, $E[U|w_2, TO] = E[Z]$ can be computed in a closed form (notice that the calculation involves simple sums of geometric sequences). Finally

$$E[U] = \sum_{w_2=1}^{w_p} E[U|w_2]\, \pi_b(w_2) \tag{16}$$

where $\pi_b(w_2)$ is derived in the previous section.

J. Proportion of Sampled Good States Terminating With NL, TD, or TO

It will be of interest to find the expected probability that an outcome of a bad state is NL, TD, or TO. These can be computed in a straightforward fashion as

$$E[P_{\rm NL}] = E[P_{\rm NL}(W_2)] = \sum_{w_2=1}^{w_p} P_{\rm NL}(w_2)\pi_b(w_2) \tag{17}$$

and similarly for $E[P_{\rm TD}]$ and $E[P_{\rm TO}]$.

[4]See Perron–Frobenius Theorem [20].

[5]In [14], an analysis of the effective bad duration was performed by 1) assuming that $\Delta(w_2) = \tau + B/\mu$ independent of w_2 and 2) assuming that when $y < \Delta(w_2)$, packet loss will be detected by a TD. Obviously, these two approximations are not true, but were made in order to arrive at a closed form solution for $E[U]$. Here, we provide a more accurate analysis that does not require either of these mentioned assumptions.

K. Average Goodput

$E[N]$ is computed from

$$E[N] = \sum_{w=1}^{w_p} E[N|W_n = w]\pi_g(w) \tag{18}$$

Hence, by application of Markov renewal-reward theory, the average goodput

$$\rho = \frac{E[N]}{E[V] + E[U]}\frac{1}{\mu}. \tag{19}$$

V. Model Validation and Discussion

In the previous section, we performed a number of approximations to the actual TCP process in order to arrive at what we called an 'idealized' process, which is semi-Markov—hence enabling the application of tools from the Markov renewal-reward theory. In this section, we evaluate the accuracy of the resulting analytical model by comparing its results against measurements from the actual TCP process, as implemented in the well-known network simulator *ns*-2 [16].

A. Our Implementation of the Channel-Driven Error Model in ns-2

The original two-state error model implemented in *ns*-2 uses packet reception as the clock for advancing the (two-state) MC representing channel state. Hence when packets are not being received (e.g., due to TO at TCP sender), the channel state is "frozen" contradicting reality. Accordingly a new two-state continuous-time Markovian model was introduced by redefining the packet dropping mechanism in the files "errmodel.cc," "errmodel.h," and "ns-errmodel.tcl"—these files and simulation script are available at [23]. A "receive" procedure is called every time a packet is scheduled for reception at the destination. In the original *ns*-2 implementation, the "receive" procedure 1) decides whether the packet is to be dropped, depending on the current state and 2) advances the two state MC in time by an amount equal to a packet transmission interval. Our implementation modifies the above procedure as follows: 1) advance the channel state from the time the last packet was processed until the time the first byte of the current packet is received (possibly involving multiple visits to good and bad states); 2) advance the channel state in time until the packet is completely received; and 3) the packet is successfully received only if the *channel remains in the good state during the entire duration of the packet reception time*, otherwise it is dropped.

In the simulation, an FTP application is used as the TCP-Reno source and a sufficiently long session is used to guarantee the steady-state is reached, i.e., simulation time is set to be at least $\max\{100t_p, 100E[X+Y]\}$. TCP-Reno parameters are set to their default values, a packet size of K (in bits) and a raw channel capacity C (in b/s) are specified in each simulation, where $\mu = C/K$. The maximum receiver advertised window was set to a value higher than w_p. We assume fine grained timers with $tcpTick$ set to 0.01.

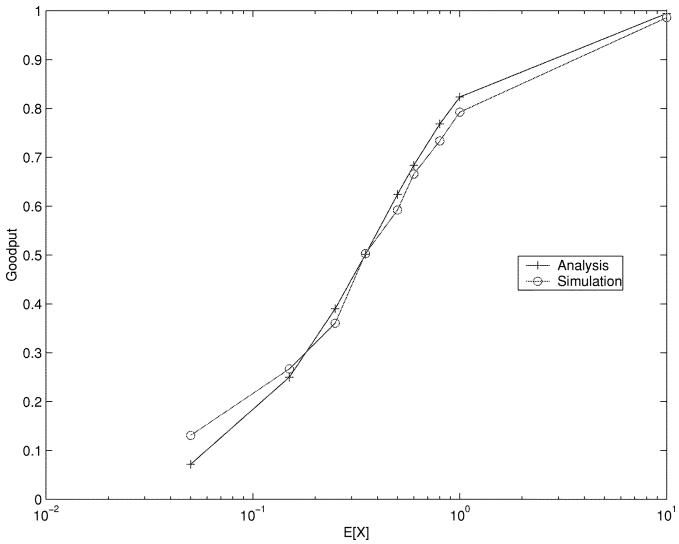


Fig. 4. Throughput comparison for the random packet loss case ($\mu = 100$ packets/s, $\beta = 1.2$, $\tau = 0.1s$, $B = 13$ packets, $w_p = 25$ packets, $K = 8 * 10^3$b).

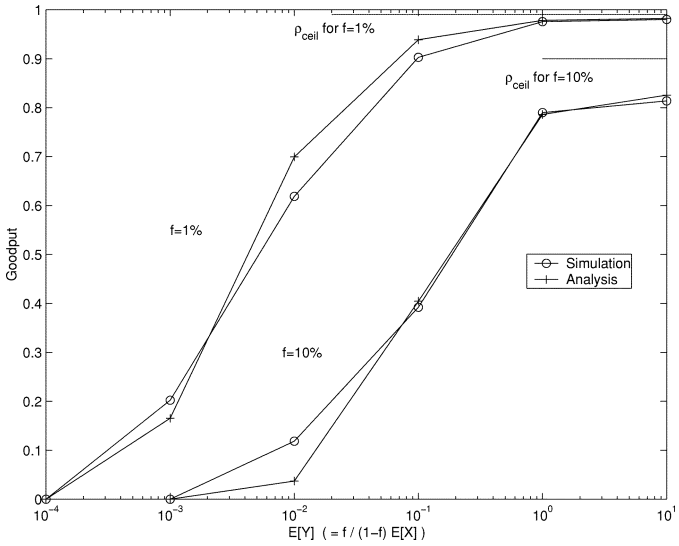


Fig. 5. Throughput comparison for the correlated packet loss case ($\mu = 100$ packets/s, $\beta = 4.0$, $\tau = 0.1s$, $B = 44$ packets, $w_p = 55$ packets, $K = 8 * 10^3$b).

B. Model Validation

Fig. 4 presents a representative comparison between the analysis results and those obtained by simulation for the i.i.d. loss case (i.e., $E[Y]$ set infinitesimally small) for $\beta > 1$. The figure shows that the model provides accurate goodput predictions for a wide range of$E[X]$. We note that for high loss rates (small $E[X]$), the analysis results slightly underestimates the goodput. This is consistent with the expected impact of the approximations 2 and 5 in Section IV-B. Further, for low loss rates, the analytical results slightly overestimates the goodput, as anticipated due to the approximation 1 in Section IV-B.

Fig. 5 contains a representative comparison of goodput for the case of correlated packet loss showing plots of the goodput against the average holding time in the bad state $E[Y]$, which equals $E[X](f)/(1-f)$. Two curves are shown for $f = 1\%$ and 10%, where on any curve, the value of f is kept fixed while

allowing $E[Y]$ and $E[X]$ to vary individually. As in the case of i.i.d. loss, we also notice that the model results slightly underestimates the goodput for the case of small $E[Y]$ (approximations 2 and 5 in Section IV-B) and slightly overestimates it for large $E[Y]$ (approximation 1 in Section IV-B). The case of $\beta < 1$ results in similar model accuracy and is omitted due to space considerations.

In order to validate the analysis for an even wider range of parameters, a total of 100 experiments were performed in which the parameters were chosen randomly as follows.

- Packet size $K \in \{0.5, 1, 1.5\}$ kB (equiprobable).
- Raw link capacity $C \in [1, 1000]$ kb/s, uniformly distributed. Thus, $\mu = C/K$. These values reflect typical values found in today's networks for wireless links capacities and TCP packet sizes.
- $5 \leq \mu T \leq 20$. The end-to-end propagation and processing delay τ is chosen such that $\mu T > 5$ since our analytical results are not accurate for smaller bandwidth-delay products due to the discretization of the window size (see Section V-D).
- $\beta = u[1.0, 2.0]$. This range represents well configured buffer sizes (ideally $\beta = 1.0$), but the analytical results are also validated for higher/smaller β (see previous results in Figs. 4 and 5).
- $E[X] = u[T, 50T]$. The average good duration is randomly chosen to span a wide range from 1 to 50 times the minimum round-trip delay T. *Notice that this range does not guarantee that assumption 2 in Section IV-B holds w.p. 1.* For example, when $E[X] = T$, the average good duration is less than the duration of one round at least $1 - e^{-1} = 63\%$ of the time. Nevertheless, we will see that the analytical results still give good results, as discussed in Section IV-B.
- $f = u[0.001, 0.30]$ The above choice represents the range of most typical wireless access links.

After selecting 100 parameter sets according to the steps above, five simulation runs (with randomly selected simulation seeds) are performed for each parameter set, during which the steady-state throughput is measured and the average of these five runs constitutes the result of each experiment which we plot against the analytical result as shown in Fig. 6. The duration of each simulation run is set long enough to guarantee steady-state operation. While there is no easy result for the appropriate run-time required for a simulation, a good heuristic (which must be checked by comparing the deviation between the different simulation runs) is $1000 \max(t_p, E[X + Y])$. Clearly, one of the major reasons for pursuing such advance modeling as in our work is the significant savings (about two orders of magnitude) vis-a-vis *ns*-2 simulations that typically took hours.

The validation figures show that the analysis matches those of the simulations with reasonable accuracy, since the points lie close to the ideal 45° line, even though we have experimented with parameter values that do not guarantee the validation of our assumptions. An analysis of the error in the 100 simulations show that in 99% (i.e., 99 of the 100 experiments), the deviation of the analysis from the simulations is less than 0.1. Further, in 84% of the experiments, the normalized error (i.e., deviation normalized to the simulation result) is less than 10%.

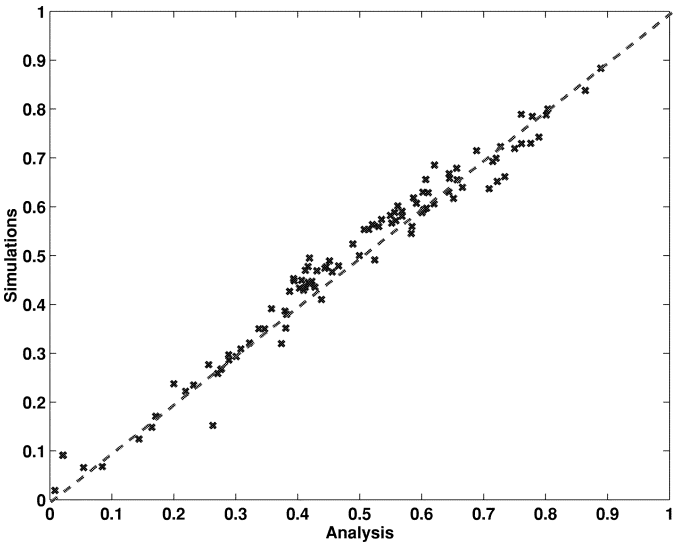


Fig. 6. Throughput from simulations versus analysis for the 100 randomly selected experiments.

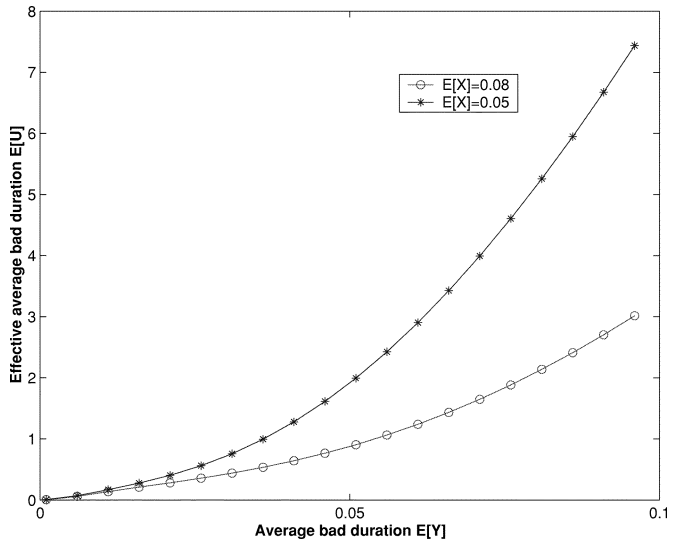


Fig. 7. The average *effective* bad duration can be many times the average value of the actual physical channel bad duration.

C. Observations

- *Fig. 7:* The behavior of the effective bad duration as a function of the actual channel bad duration is analyzed. The experiment parameters are $C = 1$ Mb/s, $\tau = 0.1$, $\beta = 1.3$ and $K = 1.5$ kB (and, hence, $\mu = C/K$). $E[Y]$ is varied from 1 to 100 ms in steps of 5 ms. Two curves are plotted for $E[X] = 50$ and 80 ms. Ideally, one would hope that the time wasted (due to idle sender or in transmission of packets that are lost) by TCP operating on top of a wireless link would not exceed that spent by the channel itself in the bad state. However, as can be seen from the figure, *the effective bad duration can be as high as 30 times the average bad duration (for $E[X] = 0.08$) and even higher (74 times the average bad duration) for smaller $E[X]$. As*

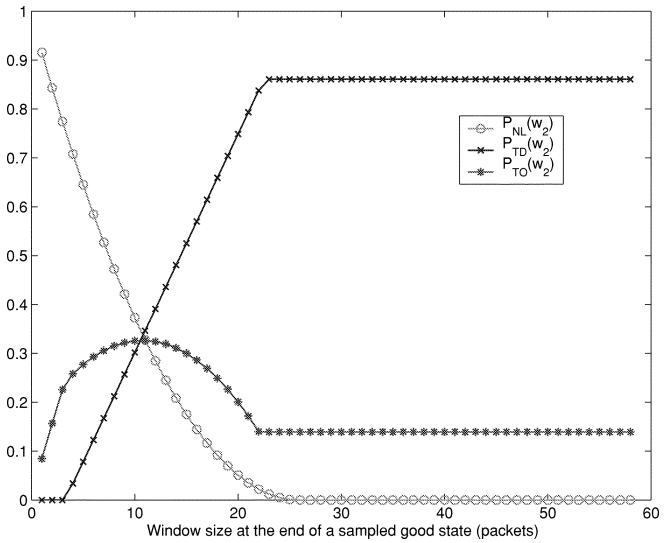


Fig. 8. The probabilities of different outcomes of a sampled good state as a function of the window size at the end of a sampled good state.

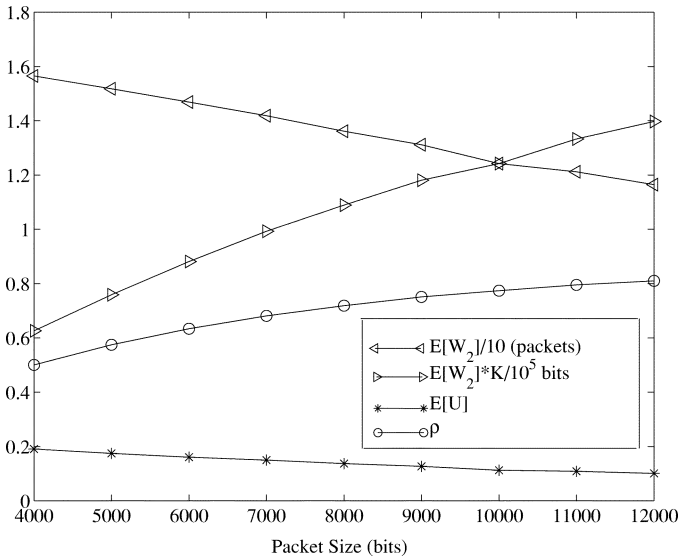


Fig. 9. The behavior of various parameters as a function of the packet size for $E[Y] = 0.01$ s and $f = 1\%$}.

stated, this factor (skipping of good states as a result of timeouts coupled with the binary exponential backoff algorithm) is a key contributor to the deterioration of TCP performance over wireless links.

- *Fig. 8:* The probability that the outcome of a sampled good state, terminating at a window size w_2, is NL, TD, or TO) [i.e., $P_{\rm NL}(w_2)$, $P_{\rm TD}(w_2)$ and $P_{\rm TO}(w_2)$] is investigated; each of these probabilities is shown as a function of $1 \leq w_2 \leq w_p$, the window size at the end of a sampled good state. We use the same parameters as in the previous figure, but with packet size 500 B, $E[X] = 0.08$, and $E[Y] = 0.001$. The results are interesting and partly counter intuitive (especially the behavior of $P_{\rm TO}(w_2)$). $P_{\rm TD}(w_2)$ increases with w_2 since larger windows increase the chances of having more than three packets transmitted following a sequence of one or more packet losses. $P_{\rm NL}(w_2)$ decreases with w_2 increasing since the idle period within a round decreases (until ultimately for $w_2 > \mu T$ it reaches zero) meaning that any bad state visit will result in at least one packet loss. The interpretation of the behavior of $P_{\rm TO}(w_2)$ requires a bit more care.
  1) Initially, (i.e., starting from $w_2 = 1$), this increases quickly with w_2 since it becomes increasingly more likely that a bad state visit incur a packet loss ($P_{\rm NL}(w_2)$ decreases) and that loss detection is likely to occur via TO rather than TD as the window size is small.
  2) As the window size increases further (beyond 4), $P_{\rm TD}(w_2)$ starts to increase, since packet loss detection using a TD becomes more likely. This causes a decrease in the slope of $P_{\rm TO}(w_2)$. Notice that $P_{\rm TO}(w_2)$ does not immediately decrease since $P_{\rm NL}(w_2)$ decreases.
  3) However, at some point ($w_2 = 10$ for this example), the decrease in$P_{\rm NL}(w_2)$ is not sufficient to prevent $P_{\rm TO}(w_2)$ from decreasing.
  4) For $w_2 \geq \mu T$, increasing the window size has negligible effect on the proportion of packet loss detection using TO or TD (note that $\mu T = 26$ in this example). This is always true if $E[Y] \ll T$. This behavior can be analytically interpreted by referring to (8). For $w_2 \geq \mu T$ and $E[Y] \ll T$, $F_Y[(w_2 - 3)/\mu] \approx 1$. Thus [from(8)], $P_{\rm TD}(w_2) \approx \bar{F}_X[3/\mu] = e^{-3/(\mu E[X])} = 0.86$ as indicated in the figure. Since $P_{\rm NL}(w_2) = 0$ for $w_2 \geq \mu T$, $P_{\rm TO}(w_2) = 1 - 0.86 = 0.14$.
- *Fig. 9* presents an analysis of the effect of TCP packet size. We show that, unlike previous results (e.g., [24]), a larger packet size always results in a higher throughput, assuming the same channel fading rate. To show this, we fix $C = 1$ Mb/s, $\tau = 0.1$ s and $\beta = 1.6$, and investigate the behavior of various parameters of interest for different choices of $E[Y]$ and$f$ (only one set of results is shown due to space limitations). In general, we have found that increasing the packet size also increases the throughput in a nonlinear fashion (unlike, for example, [7] and [8]). This can be explained by observing how other key parameters affecting the throughput vary. First, notice that the average window size at the end of a sampled good state w_2 in packets is inversely proportional to the packet size, as a direct consequence of using larger packets. However, the window size in bytes (or bits) is directly proportional to the packet size—thus, the average reward (in bits or bytes) must also be directly proportional to the packet size. Second, we observe that, the effective bad duration is inversely proportional to the packet size (assuming all other parameters remain fixed). Hence using (19), *these two factors combined result in nonlinear throughput increase as a function of the packet size*.
- *Fig. 5* shows that the goodput for channels with memory (e.g., fading channels) exhibits saturation for a fixed f as $E[Y]$ and $E[X]$ increase (this observation has been also noted in some previous works, e.g.,[24]). The reason for this is evident—when $E[X]$ is sufficiently large, TCP goodput in the good state approaches that of an ideal channel, which we denote by$\rho_{\rm ideal}$. Also, since the good states are long, all good states are sampled. Hence, for

TABLE III
THROUGHPUT FOR DIFFERENT FADE RATES

Fading	$E[X]$	$E[Y]$	Range	ρ
Fast	0.08	0.02	$D < RTT_{max}$	0.1591
Moderate	0.8	0.2	$RTT_{max} < D < t_p$	0.5367
Slow	8	2	$t_p < D < 10t_p$	0.6973
Slow	80	20	$D > 10t_p$	0.7189
Slow	800	200	$D \gg 10t_p$	0.7480

a fixed f, ρ_{ceil} for slow channel transition frequency is given by

$$\rho_{\text{ceil}} = (1 - f)\rho_{\text{ideal}} \tag{20}$$

since bad states have zero goodput.

- *Table III:* Let $D = E[X]+E[Y]$. An empirical rule for deciding whether the frequency of channel transitions, from the TCP goodput perspective, is fast or slow (i.e., fast or slow fading) is to compare D to $\text{RTT}_{\max} = (B/\mu) + \tau$ and t_p. If $D < \text{RTT}_{\max}$, this is a fast fading channel while if $D > t_p$, then we have a slow fading channel and the goodput is given by (20). An example of applying this rule is shown in Table III, which clearly shows the deleterious effects of fast fading relative to slow/moderate fading on TCP goodput for the same f. The parameters used were $\mu = 100$, $\tau = 0.01$, $K = 1$ KB and $\beta = 8.0$. Hence, $t_p = 1.215$ and $\text{RTT}_{\max} = 0.17$ and we chose$f = 0.2$ (i.e., $E[X] = 4E[Y]$).

D. Limitations and Extensions

- *Computation Complexity:* While the analysis in this paper did not lead to closed form solution of the steady-state throughput or average window size, it does provide simple expressions for the parameters needed to accurately compute them. Specifically, the elements of the transition probability matrix and reward are essentially expressed as the sum of one or two terms, each consisting of products of exponential forms. Thus, our analytical computations are much less complex than running *ns*-2 simulations. Further, the analytical results quantify some important parameters affecting TCP performance (e.g., $E[P_{\text{NL},...}]$, etc.) that would be problematic in *ns*-2 simulations (requires some nontrivial tracing to link the channel information with TCP window adaptation making the simulation even more time consuming).
- *Extensions:* The analysis in this paper was directed to TCP Reno. Generalization to other TCP versions (e.g., TCP Vegas [25]) is nontrivial, and would succeed only if a suitable renewal-reward characterization (i.e., it is possible to identify renewal instants at which the future evolution, conditioned on the current state, is independent of the past) can be found.

  Another challenging extension of the model is to the case of multiple flows. An appropriate Markovian description that accurately captures the interaction between channel and queue losses in case of a bottleneck wireless backbone link (e.g., satellite links) with multiple flows would be very useful; however the model complexity is expected to rise at least linearly with the number of flows, thus limiting the potential utility of this approach without further simplification.
- *Limitations:* The analysis in this paper assumes that the window size can be expressed in terms of "packets", and hence are integer valued. In reality, the TCP "packets" (more appropriately called "segments") are integer valued Bytes. The simulations have shown that this assumption does not have significant impact on the accuracy, mainly because we assume an infinite reservoir of fixed-size packets equal to the maximum segment size allowed (typically 500 to 1500 B). However, due to this approximation, our analysis is limited to cases where the bandwidth-delay product (μT) is larger than one. Another limitation is that we assume TCP maximum advertised window size w_m is always greater than the congestion window. In reality, the congestion window is limited by the advertised window size which may cause a throttling of the window size increase; but this can be easily incorporated in our analysis by simply replacing w_p with the maximum advertised window, and changing the analytical expressions to reflect the fact that when TCP window size reaches w_m it stays constant until there is a packet loss.

VI. CONCLUSION

In this paper, a model for TCP operation over a wireless link was presented in which the channel evolution is independent of the TCP operation. A detailed Markov renewal-reward analysis for TCPs operation is performed that accounts for both channel loss and congestion (buffer) loss. The analysis captures key TCP aspects such as triple duplicate and timeout loss detection and the binary exponential backoff algorithm. It was shown that timeouts may cause TCP to skip some good states, resulting in a longer effective bad duration. One way to alleviate this problem lies in modifications to the binary exponential backoff algorithm to allow the sampling of the channel while avoiding increasing the load on the network (in case of true congestion) which is subject of our future work. Further, based on our model, the results show that it is always recommended to use a higher packet size for a given fading rate, unlike some earlier works. The model was validated against *ns*-2 simulations for a representative number of experiments. Specifically, we explored the validity of the model for the two ranges of the normalized bandwidth-delay products (i.e., $\beta < 1$ and $\beta > 1$) and for the i.i.d., as well as correlated loss statistics. The channel-driven modeling approach provides a rule of thumb to characterize the channel fading frequency based on the number of transitions within one typical cycle.

REFERENCES

[1] J. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proc. ACM SIGCOMM'93*, 1993, pp. 289–298.
[2] K. Ramakrishnan and S. Floyd, *A Proposal to Add Explicit Congestion Notification (ECN) to IP*: IETF, Jan. 1999.
[3] A. Kumar, "Comparative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Trans. Networking*, vol. 6, pp. 485–498, 1998.
[4] A. Kumar and J. Holtzman, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link, Part II: Rayleigh Fading Mobile Radio Link," Rutgers Univ., Piscataway, NJ, Tech. Rep. WINLAB-TR-133, 1996.

[5] M. Zorzi, A. Chockalingam, and R. Rao, "Throughput analysis of TCP on channels with memory," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1289–1300, 2000.
[6] F. Anjum and L. Tassiulas, "On the behavior of different TCP algorithms over a wireless channel with correlated packet losses," in *Proc. ACM SIGMETRICS'99*, 1999, pp. 155–165.
[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Trans. Networking*, vol. 8, pp. 133–145, Apr. 2000.
[8] M. Mathis, J. Semke, J. Madhavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, 1997.
[9] J. Padhye, V. Firoiu, and D.Don Towsley, "A Stochastic Model of TCP Reno Congestion Avoidance and Control," Univ. Massachusetts, Cambridge, MA, Tech. Rep. CMPSCI-99-02, 1999.
[10] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," presented at the *Proc. ACM SIGCOMM'2000*, Stockholm, Sweden, 2000.
[11] ——, "TCP in presence of bursty losses," *Comput. Commun. Rev.*, vol. 30, no. 4, pp. 231–242, 2000.
[12] T. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. Networking*, vol. 5, pp. 336–350, 1997.
[13] M. Zorzi, R. R. Rao, and L. B Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels," in *Proc. 1995 4th IEEE Int. Conf. Universal Personal Wireless Communications*, 1995, pp. 211–215.
[14] A. Abouzeid, S. Roy, and M. Azizoglu, "Stochastic modeling of TCP over lossy links," in *Proc. INFOCOM'2000*, 2000, pp. 1724–1733.
[15] N. Chaskar, T. V. Lakshman, and U. Madhow, "TCP over wireless with link level error control: analysis and design methodology," *IEEE/ACM Trans. Networking*, vol. 7, pp. 605–615, 1999.
[16] ns-Network Simulator, 1995.
[17] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM'88*, 1988, pp. 314–329.
[18] M. Allman, V. Paxson, and W. Stevens, *TCP Congestion Control*, Apr. 1999.
[19] R. Ludwig and R. H. Katz, "The Eifel retransmission timer," *Comput. Commun. Rev.*, vol. 3, no. 3, pp. 17–27, Jan. 2000.
[20] R. Gallager, *Discrete Stochastic Processes*. Boston: Kluwer, 1996.
[21] S. Ross, *Stochastic Processes*, 2nd ed. New York: Wiley, 1996.
[22] A. Abou-Zeid, "Stochastic Models of Congestion Control in Heterogeneous Next Generation Packet Networks," Ph.D. dissertation, Dept. Elec. Eng., Univ. Washington, 2001.
[23] ——, "Stochastic Modeling of TCP/IP over Lossy Links," M.S. thesis, Dept. Elec. Eng., Univ. Washington, 1999.
[24] B. S. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, "Improving performance of TCP over wireless networks," in *Proc. 17th Int. Conf. Distributed Computing Systems*, 1997, pp. 365–73.
[25] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *IEEE J. Select. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

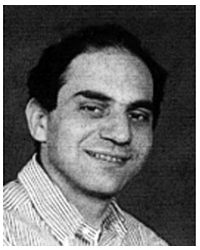
Alhussein A. Abouzeid (S'00–M'01) received the B.S. degree from Cairo University, Cairo, Egypt, in 1993 and the M.S. and Ph.D. degrees from University of Washington, Seattle, WA, in 1999 and 2001, respectively, all in electrical engineering.

He is currently an Assistant Professor in the Department of Electrical, Computer, and Systems Engineering (ECSE), Rensselaer Polytechnic Institute, Troy, NY. The industry interactions during his Ph.D. study included working with Microsoft Research, Redmond, VA (Ph.D. topic), Honeywell, Bellevue, WA (techniques for wireless access to airplanes), and at Hughes Research Labs, Malibu, CA (efficient networking protocol simulations for broadband satellite systems). He was a System Engineer with Alcatel Business Systems, Middle East Regional Office, Cairo, from 1994 to 1997 and a software engineer at the Information Technology Institute, Cairo, during 1993.

Sumit Roy (S'84–M'88–SM'00) received the B.Tech. degree from the Indian Institute of Technology, Kanpur, in 1983 and the M.S. and Ph.D. degrees from the University of California, Santa Barbara, in 1985 and 1988, respectively, all in electrical engineering, as well as the M. A. degree in statistics and applied probability in 1988.

His previous academic appointments were at the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, and at the University of Texas, San Antonio. He is presently Professor of Electrical Engineering, University of Washington, Seattle, where his research interests include analysis/design of communication systems/networks, with a topical emphasis on next-generation mobile/wireless networks. He is currently on academic leave at Intel Wireless Technology Lab working on high-speed UWB radios.

Prof. Roy is a Member of the IEEE Communications Society and is on several technical committees and TPC committees for conferences. He presently serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.



Murat Azizoglu (S'86–M'91) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1985, the M.S. degree from Ohio State University, Columbus, in 1987, and the Ph.D. degree from Massachusetts Institute of Technology, Cambridge, in 1991, all in electrical engineering.

He is the Chief Network Architect of the Transport Business Unit at Sycamore Networks, Chelmsford, MA, where he leads an architecture group developing next-generation optical networking product architectures. Prior to joining Sycamore Networks in 1999, he was a tenured Associate Professor of Electrical Engineering at the University of Washington, Seattle, during 1994–1999. From 1991 to 1994, he taught and conducted research at the George Washington University, Washington, DC, as an Assistant Professor of Electrical Engineering and Computer Science. He was a Member of Technical Staff at Bellcore (now Telcordia), Morristown, NJ, in 1989. He has published extensively in a wide range of areas including optical networks, wireless networks, communication theory, and information theory.