TCP Over Wireless with Link Level Error Control: Analysis and Design Methodology

Hemant M. Chaskar, *Member, IEEE,* T. V. Lakshman, *Senior Member, IEEE,* and U. Madhow, *Senior Member, IEEE*

***Abstract*— This paper considers the problem of supporting TCP, the Internet data transport protocol, over a lossy wireless link whose quality varies over time. In order to prevent throughput degradation, it is necessary to "hide" the losses and the time variations of the wireless link from TCP. A number of solutions to this problem have been proposed in previous studies, but their performance was studied on a purely experimental basis. This paper presents an approximate analysis, validated by computer simulations, for TCP performance over wireless links. The analysis provides the basis for a systematic approach to supporting TCP over wireless links. The specific case of a Rayleigh-faded wireless link and automatic repeat request-based link-layer recovery is considered for the purpose of illustration. The numerical results presented for this case show that a simple solution, that of using an appropriately designed link-layer error-recovery scheme, prevents excessive deterioration of TCP throughput on wireless links.**

***Index Terms*— Link-layer protocols, performance analysis, Rayleigh fading, TCP, wireless networks.**

I. Introduction

THE anticipated emergence of mobile computing depends on the ability to support data applications over heterogeneous networks comprising both wireless and wireline links. Many existing applications (including file transfers and web transactions) are based on TCP/IP, the Internet end-to-end data transport protocol. At the core of TCP is a dynamic window-based congestion control scheme proposed by Jacobson [1], which cuts back the window size (and hence, the offered rate) of the data connection upon detection of packet loss, and then gradually increases it when successful transmissions occur. When there is a wireless link in the end-to-end path of TCP connection, TCP performance suffers from significant throughput degradation due to underutilization of available wireless bandwidth [2]. This occurs because TCP cuts back its window in response to losses on the wireless link, and the subsequent growth of the window is typically sluggish, compared to variations in the quality of the wireless link.

Manuscript received July 21, 1997; revised November 2, 1998; recommended by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Pink. This work was supported by the U.S. Army Research Office under Grant DAAH04-95-1-0246 and Grant DAAG55-98-1-0219.

H. M. Chaskar was with the Coordinated Science Laboratory and the ECE Department, University of Illinois, Urbana, IL 61801 USA. He is now with Nokia Research Center, Burlington, MA 01803 USA (e-mail: hemant.chaskar@nokia.com).

T. V. Lakshman is with Bell Laboratories, Holmdel, NJ 07733 USA (e-mail: lakshman@research.bell-lab).

U. Madhow is with the Coordinated Science Laboratory and the ECE Department, University of Illinois, Urbana, IL 61801 USA (e-mail: madhow@uiuc.edu).

Publisher Item Identifier S 1063-6692(99)08191-1.

One solution to this problem is to use a suitable link-layer error-recovery mechanism that can "hide" the fluctuations of the wireless medium from TCP. Here, such an approach is referred to as "link shaping." It was shown in [3] that if TCP sees a probability of end-to-end loss above a threshold that scales as the inverse square of the bandwidth-delay product, then its throughput will deteriorate significantly. The main contribution of this paper is to show that it is possible to design a link-shaping scheme for wireless links that keeps the level of loss seen by TCP below this threshold, and to thus prevent the throughput degradation of TCP over wireless links. This is demonstrated by numerical results, based on approximate analysis validated by computer simulations, for the specific case of a Rayleigh-faded link with automatic repeat request (ARQ)-based link shaping. A more general asymptotic argument is also presented to qualitatively establish that the approach is applicable to general Markovian channel models as well.

The research reported in this paper complements the efforts of other researchers who have proposed schemes such as the split connection approach [4], the snoop protocol [5], and even link-layer recovery [6] to prevent throughput degradation of TCP over wireless links. Unlike the experimental studies cited above, the approach in this paper is to use analysis to provide a systematic framework for supporting TCP over wireless links.

The remainder of the paper is organized as follows. Section II contains the basic framework for the TCP-friendly link-shaping advocated in this paper, along with the background material on TCP dynamics and performance analysis. This framework is applied to the specific system model in Section III. Section III also describes the simulation setup used to validate the analytical throughput estimates. Numerical results are presented in Section IV. Section V contains a discussion of asymptotics for large bandwidth-delay products, and Section VI contains our conclusions.

II. Framework for TCP-Friendly Link Shaping

The framework for TCP-friendly link shaping is described in this section. Since this framework is guided by the TCP performance analysis in [3], we begin by providing some background on TCP, as well as on relevant results from [3] in Section II-A.

A. Background on TCP

TCP uses window-based flow control. At time t, the window size is denoted by $W(t)$ and is equal to the maximum

allowable number of unacknowledged packets (not counting retransmissions). For an infinite data source, the connection uses its allowable window to the fullest extent, i.e., at time t, there are indeed $W(t)$ unacknowledged packets. The window size varies dynamically in response to acknowledgments (ACK's) and detection of packet loss. The destination sends back *cumulative* ACK's: if all data segments up to $(n-1)$ have been received, then the receiver sends back the ACK "next expected $= n$." Thus, a single packet loss can be detected by consecutive ACK's having the same "next expected" number, so that TCP retransmits the packet after the number of such "duplicate ACK's" exceeds a threshold (typically three). This is the so-called "fast retransmit" option that is implemented in most versions of TCP. If packet loss is not detected in this manner, it leads to expiry of a timer. In either case, TCP-Tahoe drops its window size to one in response. Subsequently, the window grows rapidly, by one packet for every successfully acknowledged packet, until it reaches half of the window size at the last packet loss. This (typically short) phase of rapid window growth is paradoxically called "slow start," since the transmission rate during this phase is low compared to not having decreased the window at all after a loss. After slow start, the algorithm switches to "congestion avoidance," in which the window grows slowly in order to probe for extra bandwidth, by incrementing the window size by one for every window's worth of acknowledged packets. This growth continues until the maximum window size is reached, or until another packet loss is detected.

TCP-Reno is similar to TCP-Tahoe, except that it tries to avoid the slow-start phase by remaining in congestion avoidance unless there is a timer expiry. Packet loss detected via duplicate ACK's results in the window size being cut by half. If a timer expiry does occur, then the window size is reduced to one, and slow start is used to grow the window back to half its value at the time of timer expiry. See [1] for a description of TCP-Tahoe and [7], [8] for a description of TCP-Reno. See [9]–[11] for simulation studies, and [3] for an analytical characterization.

Next, some results from [3] regarding TCP performance that are relevant for the purpose here, are summarized. Consider a single TCP connection traversing an error-free bottleneck link of speed C packets/s. The bottleneck buffer is of size B packets. If the round-trip propagation delay is τ (assumed to be a constant), then the window evolution is roughly as follows: packet loss occurs when the window size reaches $W_{\rm bd}+B$, the size of the bit pipe, where $W_{\rm bd}=C\tau$ is the bandwidth-delay product. The window size drops down to one, increases exponentially during slow start to $(W_{\rm bd}+B)/2$, and then increases more slowly (almost linearly) during congestion avoidance to $W_{\rm bd}+B$, at which point there is another loss. Since the slow start phase is relatively short, the throughput is governed by the congestion avoidance phase, where the window goes from $(W_{\rm bd}+B)/2$ to $W_{\rm bd}+B$. In order to fully utilize the bottleneck link at all times, $W(t)\geq W_{\rm bd}$ must hold at all times. For $B\geq W_{\rm bd}$, this condition is satisfied throughout the congestion avoidance phase, where the bulk of the packet transfers takes place, so the link utilization is nearly 100%. Thus, the important point is that the buffering at the node must scale linearly with the bandwidth-delay product of the connection in order to achieve high link utilizations.

"Random loss" is used in [3] to model any packet losses that are not due to congestion at the bottleneck link. In particular, in the preceding scenario, suppose that a packet could be lost with probability q even after transmission on the bottleneck link, and that successive losses are independent. Upon detection of a loss, the window size, and hence the offered rate, of the TCP connection drops. The subsequent increase in window size is clocked by the arrival of ACK's, and is therefore governed by the round-trip time of the connection. Thus, if the round-trip time is large and the window size is decreased often due to random losses, the TCP connection may never operate at an offered rate that fully utilizes the available bandwidth. An approximate analysis of this phenomenon using a fixed-point argument [3] is summarized below.

If the largest window size attained during congestion avoidance "on average" is denoted by $2w_q$, then the next congestion avoidance phase starts on average at w_q. Assuming that the number of packets transmitted as the window evolves from w_q to $2w_q$ is exactly $1/q$ (this is the average number of packets transmitted successfully between successive losses), we can solve for w_q based on a continuous-time approximation to TCP dynamics during congestion avoidance. If $2w_q$ is significantly smaller than $W_{\rm bd}+B$, the maximum window size attained by TCP without random loss, then the window size during congestion avoidance, and hence the throughput, will on average be much smaller than the throughput without random loss. This leads to the following rule of thumb: if the product $qW_{\rm bd}^2$ is large (compared to 1), then random loss causes a serious deterioration in TCP throughput. In the context of this paper, this implies that the *residual* end-to-end loss probability seen by TCP after link-layer error recovery must scale inversely with the square of the bandwidth-delay product of the connection.

B. TCP-Friendly Link Shaping

Based on the preceding background, the following line of reasoning can be used to arrive at the features required for TCP-friendly link shaping.

1) Approximate analysis of TCP [3] shows that TCP throughput deteriorates if the end-to-end packet-loss probability seen by TCP is larger than $1/W_{\rm bd}^2$.
2) Consider a TCP connection that enters a wireless link through a wireline-wireless interface buffer. If the link-level error-recovery protocol is such that every packet that enters the buffer is ultimately delivered to the destination of the wireless link without TCP's own error-recovery mechanism being activated, then the only losses seen by TCP are those due to buffer overflow at the wireless link.
3) Given the packet-loss model on the wireless link, choose the size of the buffer (B) at the link, so that the buffer overflow probability is no more than $1/W_{\rm bd}^2$ when the utilization of the wireless link (or the throughput of the connection) is significant.

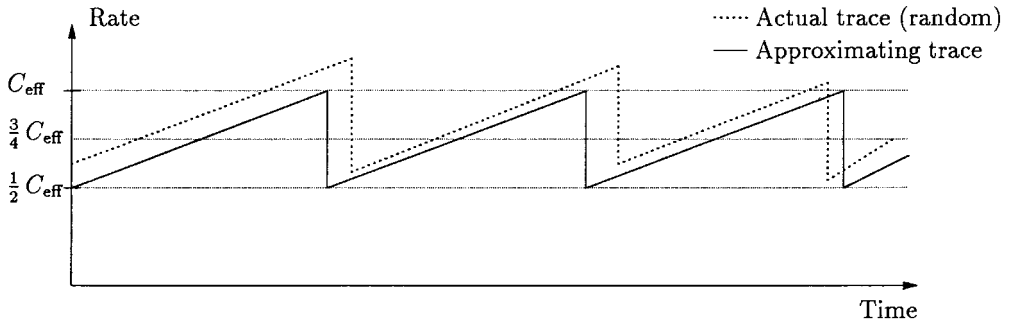


Fig. 1. Approximation to TCP window evolution.

In order not to activate TCP's own error-recovery mechanism for the packet that is still being (re)transmitted on the wireless link, the following conditions must be met. Then item 2 in the above line of reasoning is a close approximation to the actual operation of the link-layer error-recovery scheme.

1) The link layer at the destination should deliver packets to higher layers *in sequence*.
2) End-to-end round-trip timers are conservative enough to allow sufficient time for the link-layer protocol to recover from losses on the wireless link. This assumption tends to hold, in practice, for the following reasons. For a TCP connection, the source calculates the round-trip time estimate as the measured and low pass filtered round-trip time (representing the average round-trip time), plus a conservative factor proportional to the measured standard deviation [12, p. 213]. Further, in practice, timers are implemented with a coarse granularity (typically 500 ms). Finally, when a retransmission occurs, the exponential backoff strategy [12, p. 212] used in TCP timers often makes the timeout value much larger than the actual round-trip time.

With such a link shaping scheme, it is possible to estimate the throughput of the TCP connection as discussed in the next section.

C. Analytical Estimation of TCP Throughput

The arrival rate of TCP packets at the interface buffer is proportional to the TCP window size, and therefore varies as the window size fluctuates. The key feature of the analysis is computing an "effective link capacity" (denoted by C_{eff}), defined as the largest arrival rate of TCP packets at the wireless link for which the buffer overflow probability is below $1/W_{\text{bd}}^2$. Here, W_{bd} equals $C_{\text{eff}}\tau$. The interpretation of C_{eff} is as follows: while the rate offered by TCP is below C_{eff}, it sees a link that is almost as good as a wireline link, in the sense that packet loss at the interface buffer is not a dominant factor in TCP window evolution. However, if the offered rate exceeds C_{eff}, the loss rate seen by TCP due to buffer overflow becomes so high that TCP dynamics would not be able to recover from the frequent window cutbacks.

The following approximation to overall TCP throughput can now be obtained. Assume that the maximum rate attained by TCP at the end of a given congestion avoidance period is exactly C_{eff}. Then, the offered rate at the beginning of the next congestion avoidance period would be $(1/2)C_{\text{eff}}$. Because the TCP window size (and hence the offered rate) grows approximately linearly during congestion avoidance, the average rate attained during congestion avoidance is approximately $\eta = (3/4)C_{\text{eff}}$ (see Fig. 1). Ignoring the relatively short slow start phase, η can be taken to be an estimate of the TCP throughput. Note that the actual maximum rate attained by TCP during a particular congestion avoidance period may be larger or smaller than C_{eff}. However, because the buffer overflow probability is designed to be of the order of $1/W_{\text{bd}}^2$ at the arrival rate of C_{eff}, TCP will rarely achieve a maximum rate significantly larger or smaller than C_{eff}. Despite the many approximations and simplifications inherent in this estimate, this estimate provides a remarkably accurate characterization of TCP performance (see the comparison with simulations in Section IV).

To summarize, let P_a be the arrival rate of TCP packets at the wireless link normalized by the raw link capacity C. The packets are queued over the interface buffer and are transmitted until successful. Let $P_{\text{oflow}}(P_a)$ indicate the probability of buffer overflow at the interface, when the arrival rate is P_a. Then

$$C_{\text{eff}} = C \times \max\left\{P_a\colon P_{\text{oflow}}(P_a) \le \frac{1}{(P_a C\tau)^2}\right\}$$

and

$$\eta = \tfrac{3}{4}C_{\text{eff}}.$$

This analytical framework is now applied to the specific system model described below.

III. System Model

Consider a single[1] TCP connection whose source (destination) lies within a wireline network and whose destination (source) is a mobile terminal communicating with a radio port connected to the wireline network, via a wireless link. In order to concentrate on the effect of the wireless link on performance, the wireless link is assumed to be the bottleneck link and the path of the connection through the remainder of the network is simply modeled as a constant delay. The round-trip time τ of a TCP packet is defined as the time elapsing

[1] Studying a model with single connection suffices as long as there is adequate isolation between different TCP flows traversing the link. Such isolation can, for example, be achieved through the use of fair scheduling and/or buffer management algorithms at the router, as given in [13], [14].

between its release from the source and the reception of the corresponding ACK (which is generated by the destination) at the source. This time includes processing time at the source, the destination and the intermediate nodes in addition to the propagation, queueing and link-layer recovery delays, and can be substantial (tens or even hundreds of millisecond) even for small networks. For the purposes of this work, τ is used only to estimate the end-to-end loss probability that TCP can tolerate. Therefore, it suffices to use a *coarse estimate* of τ. In particular, τ is assumed to be constant in the analysis. For simplicity, it is assumed that neither link-layer nor end-to-end ACK's are lost.[2]

The link-layer error-recovery protocol transmits a packet indefinitely over the wireless link until it is successful. A link-layer window size of one is assumed, and link-layer ACK's are assumed to be instantaneous. The present framework holds even if these assumptions about the link-layer protocol are not satisfied, although detailed performance evaluation under such circumstances is beyond the scope of this work.

A. Wireless Channel Model

Consider a wireless link with raw capacity C (i.e., C is the maximum rate in TCP packets/s over the link) fully devoted to the connection of interest. This could arise, for instance, in a TDMA system with a static bandwidth assignment, or in a multiplexed system where minimum bandwidth guarantees are provided using variants of weighted fair queueing [16] or weighted Round-Robin [17] scheduling disciplines. The channel variations considered are due to Rayleigh fading, which is a widely accepted model for transmission from a base station to a mobile located in a dense scattering environment (see [18]). Wireless link outages due to handoff are not considered here. If desired, they can be incorporated as an on/off variation modulating the channel fluctuations considered here.

A distinction is made between TCP packets and packets transmitted by the link-layer protocol by calling the latter link-layer packets or "LL packets." Thus, a TCP packet may be segmented into several LL packets prior to the transmission on the wireless link, and then reassembled at the other end. Transmission over the wireless link is assumed to be time slotted, with the slot duration T being the time required to transmit an LL packet over the link. Typically, each LL packet would be encoded using an error detection/correction code. The probability of correct reception of any bit in the packet depends on the signal-to-noise ratio during the transmission of that bit. The signal-to-noise ratio varies over the duration of packet transmission due to fading. The fading model, when used in conjunction with the specific error correction/detection code used on the link, can be used to obtain the statistical models for packet losses on the link. In this work, however, modeling of packet losses due to bit errors on the wireless channel is not addressed in detail, and a worst-case packet-loss model derived from the following two-state channel model is used.

Let $S(t)$ denote the state of the wireless channel at time t. When the signal power is below a given threshold, the channel is said to be in the "bad" state $(S(t) = 0)$, otherwise it is in the "good" state $(S(t) = 1)$. The good and the bad states are assumed to last for durations which are exponentially distributed with parameters λ and μ, respectively. Thus, $\{S(t); t \in \Re\}$ is modeled as a continuous time Markov process on the state–space $\{0, 1\}$. Given this power threshold and the Doppler frequency, the mean durations of the good $(1/\lambda)$ and the bad $(1/\mu)$ states can be estimated using the level crossing analysis [18] of the fading process (See Section IV for a detailed explanation).

B. Packet-Loss Model

The probability of correct (successful) reception for a packet is governed by the channel state evolution during the time slot of its transmission. The following worst-case model is used to obtain the probability of correct reception. If the channel is in the good state at the beginning of the time slot, and remains in the good state *throughout* the duration of the time slot, then the probability of correct reception (or a "departure") is P_d. In all other cases, the probability of correct reception is assumed to be zero. If the reception is successful, the LL packet departs. Otherwise, it is retransmitted repeatedly in the following slots until a successful reception occurs.

C. Computation of C_{eff}

To determine C_{eff}, the arrival process of TCP packets at the wireless link is modeled by a stationary Bernoulli process, with the probability of TCP packet arrival in a given time slot being P_a. (If each TCP packet corresponds to n LL packets, the LL packet arrivals are in batches of n, where the batch arrivals are Bernoulli with rate P_a per time slot.) This assumption is made primarily because an exact analysis accounting for the dynamics of TCP and of the wireless channel is intractable, and its ultimate justification is through validation via simulations. However, the following comments regarding the above assumption are in order.

During the congestion avoidance phase, the actual arrival process of TCP packets at the interface buffer may be burstier than the Bernoulli process because, when the channel is in the bad state, most transmissions over the wireless link are unsuccessful, and no ACK's are generated by the receiver. When the channel state becomes good, a burst of packets gets through, thus producing a burst of ACK's. These ACK's, in turn, trigger the release of a burst of new TCP packets into the network. ACK losses and ACK compression are some of the other factors that may cause burstiness in the TCP traffic. It is possible to account for such burstiness in the arrival process using an on/off model. However, the Bernoulli assumption is used throughout the rest of this section to maintain simplicity.

A TCP packet is lost due to buffer overflow at the wireless link if any of the LL packets constituting that TCP packet cannot be accommodated in the buffer. Letting Q denote the stationary queue length at the interface buffer, and B the size

[2] Since TCP uses cumulative ACK's, the only consequence of ACK losses is increased burstiness in the traffic on the forward path [15]. One way to account for this burstiness is stated in Section III-C.

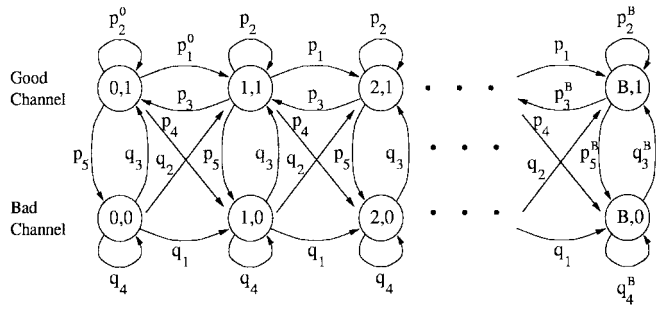


Fig. 2. Transition diagram for Markov chain (N_k, S_k).

of the buffer, the stationary probability of such a loss event is given by Prob $[Q \geq B - n + 1]$. Thus, for a given target throughput of $\eta = (3/4)C_{\text{eff}}$, the objective is to find B such that

$$\text{Prob}\,[Q \geq B - n + 1] \leq \frac{1}{W_{\text{bd}}^2}, \quad \text{when } C_{\text{eff}} = P_a C.$$

1) Buffer Overflow Probability for Large LL Packets: The case $n = 1$ is first analyzed in detail. The convention is that an arrival occurs just before the end of the time slot; a wireless transmission begins at the beginning of the time slot if the buffer is nonempty; and if the transmission is successful, departure occurs at the end of the time slot. Let N_k and S_k denote the number of LL packets in the buffer and the channel state (good or bad), respectively, just after the beginning of the kth time slot. Then, $\{(N_k, S_k), k \geq 1\}$ is a Markov chain on the state–space $\{(m, s)$: $0 \leq m \leq B, s \in \{0, 1\}\}$. Its transition probability diagram is shown in Fig. 2. The transition probabilities are defined as follows:

$p_1 = P_a(1 - P_d)e^{-\lambda T} + P_a P_{11}$
$p_2 = P_a P_d e^{-\lambda T} + (1 - P_a)\,(1 - P_d)e^{-\lambda T} + (1 - P_a)P_{11}$
$p_3 = (1 - P_a)P_d e^{-\lambda T}$
$p_4 = P_a P_{10}$
$p_5 = (1 - P_a)P_{10}$
$p_1^0 = P_a(e^{-\lambda T} + P_{11})$
$p_2^0 = (1 - P_a)(e^{-\lambda T} + P_{11})$
$p_2^B = (1 - P_d)e^{-\lambda T} + P_{11}$
$p_3^B = P_d e^{-\lambda T}$
$p_5^B = P_{10}$
$q_1 = P_a P_{00}$
$q_2 = P_a P_{01}$
$q_3 = (1 - P_a)P_{01}$
$q_4 = (1 - P_a)P_{00}$
$q_3^B = P_{01}$
$q_4^B = P_{00}$

where

$P_{11} = \text{Prob}\,[S_{k+1} = 1 | S_k = 1] - e^{-\lambda T}$
$P_{ij} = \text{Prob}\,[S_{k+1} = i | S_k = j]$, for $0 \leq i, j \leq 1$ and $(i, j) \neq (1, 1)$.

The probability P_{11} is thus the probability of the channel being in the the good state at the end of the time slot, given that it is in the good state at the beginning of the time slot, but entering the bad state at some point in between. For all other (i, j), P_{ij} denotes the probability that the slot ends in state j, given that it starts in state i. To understand how the transition probabilities are determined, consider p_1 as an example. If the slot begins with the channel in the good state, and the buffer is neither empty nor full, then the queue size increases by one, *and* the next slot also begins with the channel in the good state, if either of the following two mutually exclusive events occur.

1) The channel remains in the good state throughout the transmission slot, but the transmission is unsuccessful and an arrival occurs.
2) The channel enters the bad state at some point during the transmission slot (so that the transmission is unsuccessful), but the slot ends with the channel in the good state and an arrival occurs.

The probability of event 1 is $e^{-\lambda T}(1 - P_d)P_a$, the probability of event 2 is $P_{11}P_a$, and p_1 is given by the sum of these probabilities.

Now let $\overline{\pi} = (\overline{x}_0 \overline{x}_1 \cdots \overline{x}_B)$ denote the stationary probability distribution for (N_k, S_k), where $\overline{x}_i$ is a row vector with entries (Prob $[N_k = i, S_k = 0]$, Prob $[N_k = i, S_k = 1]$). The probability of TCP packet loss is given by $P_B = x_B(0) + x_B(1)$. From Fig. 2, $x_0(0)$ and $x_0(1)$ can be related as

$$x_0(0) = \frac{p_5}{q_1 + q_2 + q_3} x_0(1). \tag{1}$$

Identifying the vector $\overline{\pi}$ as a matrix-geometric probability vector [19] yields that

$$\begin{aligned} \overline{x}_1' &= A_1 \overline{x}_0' \\ \overline{x}_k' &= A_2 \overline{x}_{k-1}', \quad \text{for } 1 < k < B \\ \overline{x}_B' &= A_3 \overline{x}_{B-1}' \end{aligned}$$

where the elements of the matrices A_1, A_2 and A_3 are such that the following relations hold:

$$\begin{aligned} x_1(0) &= \frac{q_1 p_3 + (q_1 + q_2)p_5}{p_3(q_1 + q_2 + q_3)} x_0(0) \\ &\quad + \frac{p_4 p_3 + (p_1^0 + p_4)p_5}{p_3(q_1 + q_2 + q_3)} x_0(1) \\ x_1(1) &= \frac{q_1 + q_2}{p_3} x_0(0) + \frac{p_1^0 + p_4}{p_3} x_0(1). \end{aligned}$$

For $1 < i < B$

$$\begin{aligned} x_i(0) &= \frac{q_1 p_3 + (q_1 + q_2)p_5}{p_3(q_1 + q_2 + q_3)} x_{i-1}(0) \\ &\quad + \frac{p_4 p_3 + (p_1 + p_4)p_5}{p_3(q_1 + q_2 + q_3)} x_{i-1}(1) \\ x_i(1) &= \frac{q_1 + q_2}{p_3} x_{i-1}(0) + \frac{p_1 + p_4}{p_3} x_{i-1}(1). \end{aligned}$$

Finally

$$\begin{aligned} x_B(0) &= \frac{p_4 p_3^B + (p_1 + p_4)p_5^B}{p_3^B q_3^B} x_{B-1}(1) \\ &\quad + \frac{q_1 p_3^B + (q_1 + q_2)p_5^B}{p_3^B q_3^B} x_{B-1}(0) \\ x_B(1) &= \frac{p_1 + p_4}{p_3^B} x_{B-1}(1) + \frac{q_1 + q_2}{p_3^B} x_{B-1}(0). \end{aligned} \tag{2}$$

An efficient algorithm for determining the stationary probability distribution can be obtained from (1) and (2). Starting with an arbitrary nonzero value for $x_0(1)$, (1) and (2) express

the stationary probabilities of all the states in terms of $x_0(1)$. The actual values of the stationary probabilities of the states are then found by normalization.

2) Buffer Overflow Probability for Small LL Packets: In this case, the process $\{(N_k, S_k), k \geq 1\}$ is again a Markov chain on the state–space $\{(m,s)\colon 0 \leq m \leq B, s \in \{0,1\}\}$. Because N_k can increase by $n>1$ in a single transition, the preceding is no longer a quasi-birth-death process. However, the fact that N_k cannot decrease by more than one (skip free to the left) allows efficient evaluation of the stationary probability distribution by standard matrix-geometric methods [19].

In this case, if the states are arranged as $[(0,0),(0,1),(1,0),(1,1),\cdots,(B,0),(B,1)]$, then the transition probability matrix has the form found at the bottom of the page, where

$$A_0^0 = \begin{pmatrix} q_4 & q_3 \\ p_5 & p_2^0 \end{pmatrix}, \quad A_0 = \begin{pmatrix} q_4 & q_3 \\ p_5 & r_2 \end{pmatrix}$$
$$A_1 = \begin{pmatrix} q_1 & q_2 \\ p_4 & p_1^0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 \\ 0 & p_3 \end{pmatrix}$$
$$A_3 = \begin{pmatrix} 0 & 0 \\ 0 & r_1 \end{pmatrix}, \quad A_4 = \begin{pmatrix} q_1 & q_2 \\ p_4 & p_1 \end{pmatrix}$$
$$A_0^B = \begin{pmatrix} q_4^B & q_3^B \\ p_5^B & p_2^B \end{pmatrix}, \quad A_2^B = \begin{pmatrix} 0 & 0 \\ 0 & p_3^B \end{pmatrix}.$$

Here $r_1 = P_a P_d e^{-\lambda T}$ and $r_2 = p_2 - r_1$. An outline of the algorithm employed to compute the stationary probability distribution for this case is given below. Referring to the pair of states $\{(m,0),(m,1)\}$ as "phase" m, where $0 \leq m \leq B$, let $\pi(m,s)$ indicate the stationary probability of the state (m,s), where $s = 0, 1$. Assuming that $c\pi(k,0)$ and $c\pi(k,1)$ are known for all $k \leq m$ (where c is an arbitrary constant determined after normalization), $c\pi(m+1,1)$ is computed by equating the probability flows across the boundary between phases m and $m+1$. The value of $c\pi(m+1,0)$ can then be found by equating the probability flow out of and into the state $(m+1,1)$. After propagating this computation from $m=0$ to $m=B$, the actual values of the stationary probabilities are obtained by normalization. The probability of TCP packet loss is then given by $\Sigma_{m=B-n+1}^{B}\,[\pi(m,0)+\pi(m,1)]$.

The analytical throughput estimates obtained for this model are compared against computer simulations. The details of this simulation are given in the next section. Prior to that, it is worth listing a number of approximations (in addition to the approximation for TCP window evolution and the Bernoulli modeling of TCP traffic) that are implicit in our analysis.

1) The analysis does not consider the reduction in the net throughput due to packets *retransmitted* by TCP. This is a minor point, because the latter is a rare event (occurring due to buffer overflow, which is designed to have probability of the order of $1/W_{\text{bd}}^2$). On the other hand, link level retransmissions, which are much more common, are accounted for in detail.
2) The relatively short slow start phase is ignored, because it does not contribute much to the TCP throughput.
3) Expiry of a coarse TCP timer when the fast retransmit option fails, which could lead to idle periods of 500 ms or more, is not accounted for in the analysis. This does not affect the performance estimates much for TCP-Tahoe because its coarse timer expiry is infrequent (occurring typically when a packet loss is detected via fast retransmit and the packet is retransmitted, but the retransmission is also lost due to congestion at the interface buffer). For TCP-Reno, multiple packet losses often cause coarse timer expiry, which is one reason for its performance being so much worse (see Section IV) than that of TCP-Tahoe (and the analytical estimate).

D. Simulation Setup

The block diagram of the simulation setup is shown in Fig. 3. Simulations are run with the Tahoe and the Reno versions of TCP. The Tahoe version includes the fast retransmit feature with a duplicate ACK threshold of three [8]. The simulation uses coarse-grained timers with a granularity of 500 ms, which is a typical value used in many systems. Attention is restricted to a single TCP-controlled source that always has data to send, so that only the TCP window limits the number of packets released into the network. The source always sends packets of 576 B, a typical value in use. The TCP destination sends an ACK for every received packet. Other ACK options in use (such as delayed ACK's [12, p. 225] or sending an ACK for every other received packet) are not simulated. ACK packets are 40 B long and do not encounter any queueing.

It is assumed that the links from the source to the wireless interface and those over which ACK's travel from the destination to the source both introduce negligible loss and delay variations. Thus, the round-trip delay due to these links is assumed to be constant. Variations in quality of the wireless channel are examined using a Markovian simulator and also the Jakes' fading simulator [18], [20] (see numerical results in

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \cdot \\ B-n \\ B-n+1 \\ B-n+2 \\ \cdot \\ B \end{array}
\begin{array}{c}
\begin{array}{ccccccccccccc} 0 & 1 & 2 & \cdot & n & n+1 & n+2 & \cdot & B-n & B-n+1 & \cdot & B-1 & B \end{array} \\
\begin{pmatrix}
A_0^0 & 0 & 0 & \cdot & A_1 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 \\
A_2 & A_0 & 0 & \cdot & A_3 & A_4 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 \\
0 & A_2 & A_0 & \cdot & 0 & A_3 & A_4 & \cdot & 0 & 0 & \cdot & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & \cdot & 0 & 0 & 0 & \cdot & A_0 & 0 & \cdot & A_3 & A_4 \\
0 & 0 & 0 & \cdot & 0 & 0 & 0 & \cdot & A_2^B & A_0^B & \cdot & 0 & 0 \\
0 & 0 & 0 & \cdot & 0 & 0 & 0 & \cdot & 0 & A_2^B & \cdot & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & \cdot & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & A_2^B & A_0^B
\end{pmatrix}
\end{array}$$

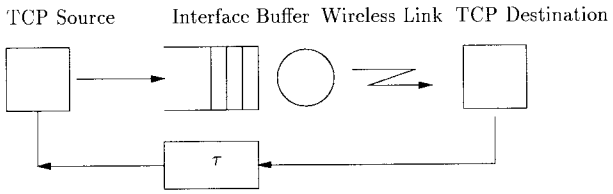


Fig. 3. Block diagram of the simulation setup.

Section IV). The packet-loss model of Section III-B is used to determine the packets received in error.

Two systems are simulated: System 1 uses 576 B LL packets (the same as TCP packets) and System 2 uses 53 B (ATM cell) LL packets. In System 2, each ATM cell is taken to have 48 B of payload and 5 B of ATM header. Hence, each TCP packet converts to 12 ATM cells. In System 2, a TCP packet arriving at the wireless link is dropped entirely if the buffer cannot hold any of the 12 LL packets that constitute the TCP packet.

In the next section, the throughput estimates obtained using analysis are compared with the simulation results for some representative parameter choices.

IV. Numerical Results

Consider a TCP connection traversing a wireless link with a raw capacity of 1 Mb/s. For a 576 B TCP packet, this translates to $C \approx 217$ TCP packets/s. Two different values for the round-trip delay τ are considered: 40 and 400 ms. The net round-trip delay is the sum of τ and the delay due to queueing and retransmission on the wireless link. The latter is not included in the value of $W_{\rm bd}$ used to determine the tolerable loss probability $1/W_{\rm bd}^2$, because the purpose is only to obtain an estimate of the right order of magnitude. For $\tau = 40$ ms, the bandwidth-delay product $C\tau$ is about 8.7 packets, so that the tolerable end-to-end packet-loss probability is about 10^{-2}. Similarly, for $\tau = 400$ ms, the bandwidth-delay product and the desired end-to-end loss probability are about 100 and 10^{-4}, respectively.

Three different channels, corresponding to the Doppler frequencies of 10, 30, and 100 Hz, are considered. The bad state corresponds to a signal power 10 dB below the nominal value. The error probability for an LL packet in the good state is assumed to be 0.01.

Two sets of results are presented. In the first set of results, the mean durations of the good and the bad states for the above values of Doppler frequencies are read off from plots in [18], as follows.[3] From [18, Fig. 1.3-5], the mean duration of a fade which is 10 dB below the nominal signal power is seen to be approximately $0.1/f_d$, where f_d is the Doppler frequency. This is the mean duration of the bad state in the present context. On the other hand, the upward crossings of a power level 10 dB below nominal occur at a rate approximately f_d (see [18, Fig. 1.3-4]). Thus, $1/f_d$ is the sum of the mean durations of successive good and bad states. The mean duration of the good state can therefore be closely approximated by $(1 - 0.1/f_d) \approx (1/f_d)$. Thus, the analysis is carried out with a Markovian channel model with $1/f_d$ and $0.1/f_d$ as the mean durations of the good and the bad states, respectively. The analytical throughput estimates are compared against simulations for the same Markovian channel model. This comparison therefore checks whether the crude description of TCP behavior employed by the analysis can provide good performance estimates.

[3] More exact values can be computed from the power spectrum of the fading random process, but this approach suffices to obtain performance estimates in the present context.

TABLE I
System 1, 400 MS Round-Trip Delay, Markovian Simulator

Doppler (Hz)	Mean duration of good(bad) state (ms)	Percent link utilization Analysis//Tahoe simulation/Reno simulation	
		B=8 TCP-packets	B=16 TCP-packets
10	100(10)	20.70//24.81/23.60	43.87//41.99/34.01
30	30(3)	30.98//27.08/26.22	47.02//40.02/31.53
100	10(1)	20.55//22.08/20.86	32.10//30.99/25.80

TABLE II
System 1, 400 MS Round-Trip Delay Jakes' Fading Simulator

Doppler (Hz)	Percent link utilization Analysis//Tahoe simulation/Reno simulation		
	B=8 TCP-packets	B=12 TCP-packets	B=16 TCP-packets
10	22.12//29.82/27.65	36.52//38.13/34.51	46.57//44.45/38.56
30	33.97//32.39/30.95	45.67//37.26/34.82	51.00//44.17/38.62
100	23.25//26.51/25.84	31.27//32.69/29.08	35.47//36.89/31.83

TABLE III
System 1, 8 TCP-Packet Buffer, Jakes' Fading Simulator

Doppler (Hz)	Mean duration of good(bad) state (ms)	Percent link utilization			
		τ=40 ms		τ=400 ms	
		Analysis	Simulation (Tahoe/Reno)	Analysis	Simulation (Tahoe/Reno)
10	115.11(9.8)	54.90	55.74/38.82	22.12	29.82/27.65
30	40.12(3.36)	54.83	53.41/42.07	33.97	32.39/30.95
100	11.52(0.98)	37.80	43.84/33.80	23.25	26.51/25.84

In the second set, the simulations are run with the Jakes' fading simulator described in [18, pp. 70–76]. For the analysis (which still assumes a Markovian channel), the mean durations of the good and the bad states are computed from samples of the random process generated by the Jakes' fading simulator. In the simulation, an LL packet is lost if the power of the process generated by the simulator drops 10 dB below the nominal power anytime during the packet transmission slot. The second set of results checks the accuracy of the Markovian channel model, in addition to the modeling of TCP.

Two systems are considered for performance evaluation:

- *System 1:* Here, an LL packet is the standard TCP packet. The use of long packets in such a situation may be suboptimal, but avoids the need for segmentation and reassembly of TCP packets at the link layer. Numerical results for System 1 are displayed in Tables I–III and Fig. 4.
- *System 2:* The LL packet is assumed to be an ATM cell, so that a 576 B TCP packet is segmented into

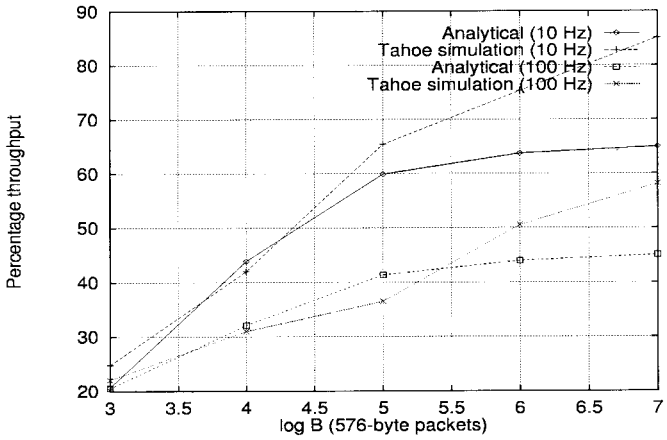


Fig. 4 System 1, 400-ms round-trip delay Markovian simulator.

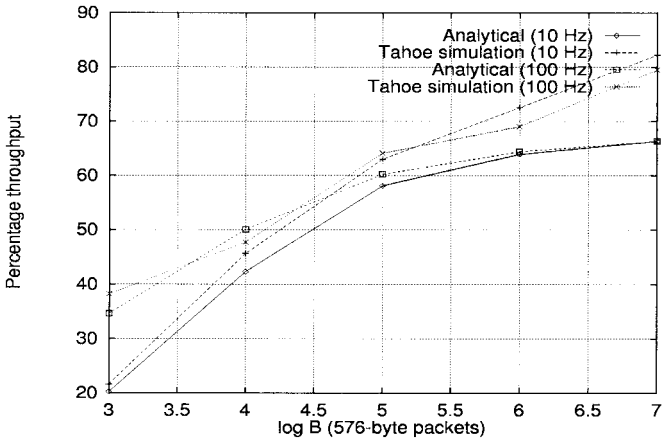


Fig. 5 System 2, 400-ms round-trip delay Markovian simulator.

TABLE IV
SYSEM 2, 400 MS ROUND-TRIP DELAY, MARKOVIAN SIMULATOR

Doppler (Hz)	Mean duration of good(bad) state (ms)	Percent link utilization Analysis//Tahoe simulation/Reno simulation	
		B=96 cells	B=192 cells
10	100(10)	20.25//21.67/17.98	42.30//45.61/39.92
30	30(3)	32.85//35.38/33.74	50.13//48.66/42.85
100	10(1)	34.65//38.22/35.03	50.04//47.71/42.77

TABLE V
SYSTEM 2, 400 MS ROUND-TRIP DELAY, JAKES' FADING SIMULATOR

Doppler (Hz)	Percent link utilization Analysis//Tahoe simulation/Reno simulation		
	B=96 cells	B=144 cells	B=192 cells
10	21.33//30.20/27.83	34.65//41.65/39.67	44.01//48.31/43.88
30	33.30//39.39/37.16	44.91//44.71/44.02	51.03//50.27/46.27
100	35.55//41.25/38.48	45.81//44.18/44.01	51.12//50.02/46.36

TABLE VI
SYSTEM 2, 96-CELL BUFFER, JAKES' FADING SIMULATOR

Doppler (Hz)	Mean duration of good(bad) State (ms)	Percent link utilization			
		τ=40 ms		τ=400 ms	
		Analysis	Simulation (Tahoe/Reno)	Analysis	Simulation (Tahoe/Reno)
10	115.11(9.8)	49.77	61.99/33.14	21.33	30.20/27.83
30	40.12(3.36)	54.45	62.62/37.31	33.30	39.39/37.16
100	11.52(0.98)	54.00	61.94/37.46	35.55	41.25/38.48

$n = 12$ cells (assuming that 48 out of the 53 B's of an ATM cell correspond to data) at the link layer. Each cell is independently transmitted over the wireless channel. The numerical results for this system are given in Tables IV–VI and Fig. 5.

Comments on Numerical Results: The agreement between analysis and simulation is not exact, which is to be expected given the number of approximations involved. However, as seen from Tables I and IV, the results of the analysis are very close to those obtained from simulations of TCP-Tahoe for the two-state Markovian channel models. Even when compared to simulations for a Rayleigh fading channel using the Jakes' simulator (Tables II, III, V, and VI), the accuracy of the analytical estimates is acceptable, even though the analysis employs a Markovian approximation for a channel simulated using the (non-Markovian) Jakes' simulator.

It is seen from Figs. 4 and 5, that the analytical throughput estimates are pessimistic when the actual throughput is close to the (good) output rate of the wireless link. The reason is the following: analytical throughput estimate is obtained as $(3/4)C_{\text{eff}}$, and hence, cannot exceed 75% of the (good) output rate. When C_{eff} is close to the (good) output rate (which occurs due to the provision of large buffer), even if the TCP window drops to half its size at the time of packet loss, this reduced window size is still large enough to keep the *rate* of packet arrivals equal to its value at the time of loss. In other words, the proportional relationship between the rate of packet arrivals and the size of the TCP window does not hold in that regime. Since the analysis assumes such a proportional relationship, it tends to be conservative.

In all cases considered, TCP-Reno gives much poorer performance than TCP-Tahoe (see Section VI for further discussion of this point). Also, in all these cases, the throughput without link-level error recovery was found to be nearly zero; hence, these results are not reported in detail.

For a fixed LL packet length, smaller throughput is expected for fast fading because a given packet is more likely to encounter a bad channel state during its transmission slot. However, slow fading can also cause poor performance, since the bad channel state persists for a longer time, during which the interface buffer accumulates packets, thereby increasing the probability of buffer overflow. These observations are reflected in Table I. There, for a buffer size of 8 TCP packets, the performance with moderate fading is the best. As the buffer size is increased to 16 TCP-packets, the effect of packet accumulation during the bad state becomes less significant; thus, the effect of a packet encountering the bad state during its transmission period becomes dominant. The consequence

is that the throughput for slow and moderate fading is higher than that for fast fading.

In fast fading environments, the use of large LL packets degrades performance, because a longer packet is more likely to encounter a bad state during its transmission. These predictions are seen to hold: comparing the results in Tables II and V, System 2 gives higher throughput for a Doppler frequency of 100 Hz than does System 1. This occurs even though the comparison is biased in favor of System 1: because the LL packet size in System 1 is larger, there is an implicit assumption of a smaller probability of TCP packet loss in the good state for System 1, as the probability of LL packet loss during the good state is taken to be the same in both systems.

While the preceding results were obtained for the specific case of Rayleigh faded wireless link, the following qualitative argument shows that, in general, for links with Markovian packet-loss statistics, the link-layer approach is expected to satisfactorily hide wireless losses from TCP with an interface buffer of moderate size.

V. Asymptotics for Large Bandwidth-Delay Products

In this section, it is argued that, even for lossless (wireline) links, the buffering required at the bottleneck link must scale linearly with the bandwidth-delay product as the latter becomes larger, while the buffering required to hide Markovian link quality fluctuations from TCP need only scale logarithmically with the bandwidth-delay product. The reader is cautioned against concluding that smaller buffers are required at the wireless links, as compared to those at the wireline links. The argument merely implies that, for TCP, the dynamic window algorithm itself is more dominant than the link shaping in terms of dictating the buffer requirements.

A. Wireline Networks/Lossless Links

Consider a bottleneck link that has maximum output speed of μ packets/s. In order to concentrate on the bottleneck link, the path of data packets and ACK's through the rest of the network is modeled as constant delay τ. Thus, the size of the end-to-end packet pipe excluding the buffering at the bottleneck link is $\mu\tau$. Let $f(\mu\tau)$ denote the buffering provided at the bottleneck link, which is the function of $\mu\tau$. Assume that $f(\mu\tau)$ scales sublinearly with $\mu\tau$, i.e.,

$$\lim_{\mu\tau\to\infty} \frac{f(\mu\tau)}{\mu\tau} = 0.$$

It has been established in [3] that in the case of TCP-Tahoe, $f(\mu\tau) > (1/3)(\mu\tau)$ is required in order to avoid "double slow start." However, if $f(\mu\tau)$ scales sublinearly with $\mu\tau$, this condition will be violated for a large-enough $\mu\tau$ and, hence, a double slow start will occur. Thus, while studying the asymptotics for TCP-Tahoe in such a case, a double slow start must be accounted for; hence, the congestion avoidance is assumed to always start from a window of size one. This phenomenon is absent in TCP-Reno (unless the dropping of the window is due to a timeout, which is a rare event). The case of TCP-Tahoe is considered first.

Define the total size of the packet pipe to be $W_m = \mu\tau + f(\mu\tau)$. Let b_1 denote the first batch of one packet, b_2 the next batch of two packets, b_3 the batch after that of three packets, and so on. Then, starting with a window of size one and a congestion avoidance threshold of one, the amounts of time required to complete[4] the transmissions of batches b_1, b_2, b_3, and so on up to $b_{\mu\tau}$ are $\tau, \tau + (1/\mu)$, $\tau + (1/\mu), \cdots$, and $\tau + (1/\mu)$, respectively. At this time, the window size increases to $\mu\tau + 1$, and the bottleneck queue begins to build up. Thus, the amounts of time required to complete the transmissions of batches $b_{\mu\tau+1}, b_{\mu\tau+2}$, and so on up to b_{W_m} are $\tau + (1/\mu), \tau + (2/\mu)$, $\tau + (3/\mu), \cdots$, and $\tau + (f(\mu\tau)/\mu)$, respectively. At this point, the window size increases to $W_m + 1$ and loss occurs, causing the window size to drop to one. Hence, the throughput per congestion avoidance phase (defined as the total number of packets transmitted in the congestion avoidance phase divided by the total duration of the congestion avoidance phase) is given by the equation shown at the bottom of the page. Thus, $\lim_{\mu\tau\to\infty} \eta_{\text{Tahoe}}(\mu\tau) = (\mu/2)$.

Along similar lines, it can be shown that if timeouts are rare, $\lim_{\mu\tau\to\infty} \eta_{\text{Reno}}(\mu\tau) = (3/4)\mu$.

This establishes that, even in the absence of random loss, in order to achieve high utilizations (more than 50% in case of TCP-Tahoe and more than 75% in case of TCP-Reno), the buffering at the bottleneck must scale linearly with the bandwidth-delay product of the connection.

B. Lossy Links with Markovian Packet-Loss Statistics

Recall that C_{eff} is defined as the maximum input rate at which the buffer overflow probability, for buffer size B, is $1/W_{\text{bd}}^2$ (which is the end-to-end loss probability TCP can tolerate). Consider a general Markovian model for packet losses on the link. For a Rayleigh fading link, such a model could arise from a refinement of the two-state model considered here, so as to model the amplitude variations on the channel more closely [21]–[23]. In addition, because the traffic is Markovian

[4] Transmission of a packet is said to be complete when the acknowledgment for that packet is received at the source.

$$\eta_{\text{Tahoe}}(\mu\tau) = \frac{1+2+3+\cdots+W_m}{\tau + \sum_{i=2}^{\mu\tau}\left(\tau + \frac{1}{\mu}\right) + \sum_{i=1}^{f(\mu\tau)}\left(\tau + \frac{i}{\mu}\right)}$$

$$= \mu \frac{\frac{1}{2}\,\frac{W_m}{\mu\tau}\,\frac{(W_m+1)}{\mu\tau}}{\frac{1}{\mu\tau} + \frac{(\mu\tau-1)}{\mu\tau}\,\frac{(\mu\tau+1)}{\mu\tau} + \frac{f(\mu\tau)}{\mu\tau} + \frac{1}{2}\,\frac{f(\mu\tau)}{\mu\tau}\,\frac{(f(\mu\tau)+1)}{\mu\tau}}.$$

(due to the Bernoulli assumption or even if the traffic is modeled to reflect the burstiness of ACK's that in turn depends on the channel model), the tail of the stationary queue-length distribution is exponentially bounded[5] [24], i.e.,

$$\lim_{B\to\infty} \frac{1}{B} \log \text{ Prob } [Q > B] = -\beta < 0.$$

In other words, Prob [overflow] $\approx e^{-\beta B}$. For the desired overflow probability of $1/W_{\text{bd}}^2$, this yields that the required buffer size is $B \approx (2/\beta) \log W_{\text{bd}}$. Note that β is the function of the channel and the traffic characteristics and the value of C_{eff} being sought. As the link utilization is increased by increasing C_{eff}, the decay factor β decreases.

VI. Conclusion

The approximate performance analysis of TCP presented here, together with the notion of effective link capacity, provides the basis for a systematic approach to supporting TCP over wireless links. Numerical results presented in Section IV show the effectiveness of link-layer error recovery (also observed experimentally in [6]) to achieve high link utilizations with TCP-Tahoe on wireless links. They also validate the analytical framework for TCP-friendly link shaping, provided in this paper. Although the specific case of Rayleigh faded wireless link with ARQ-based link shaping is addressed, the framework also applies to other kinds of lossy links and link-layer recovery mechanisms. (The method of estimating the residual packet-loss probability at the lossy link may differ from case to case.) The asymptotic argument presented in Section V lends further support to these claims.

The performance of TCP-Reno is invariably worse than that of TCP-Tahoe. This is mainly because of the lack of robustness of TCP-Reno against multiple closely spaced losses. TCP-Reno cuts its window by half for *each* loss detected, resulting in multiple window cutbacks for the same congestion episode. In contrast, TCP-Tahoe cuts its window to one regardless of the number of losses, and then grows it back rapidly during slow start to half the value of the window at which the *first* loss occurred. Thus, while TCP-Reno tries to achieve higher throughput by avoiding the slow start phase, TCP-Tahoe typically operates with a *higher* window size during the congestion avoidance phase, in which the bulk of packet transfers take place. Further, the fast retransmit option, while effective in identifying that at least one loss has occurred (as required by TCP-Tahoe), fails to identify multiple losses, as required for proper operation of TCP-Reno. This can lead to an idle period due to a coarse timeout, which is another reason for the poorer throughput attained by TCP-Reno.

Throughout this paper, attention was restricted to a single TCP connection. When multiple connections share the bottleneck link, fair queueing and/or appropriate buffer management can be used to provide isolation among different connections, as proposed in [13], [14]. Thus, given the resources (the bandwidth and the buffer) allocated to TCP connection at the bottleneck, the analytical approach presented here enables the estimation of realizable throughput. Conversely, QoS (throughput) negotiations can be done on the basis of available resources at the bottleneck link. A TCP connection seeking a certain throughput can be admitted at the bottleneck link if adequate resources are available to satisfy the $1/W_{\text{bd}}^2$ criterion at that throughput.

[5]The effective bandwidth framework considers a Markovian source and a deterministic channel, but the analysis extends immediately to accommodate a Markovian channel as well.

References

[1] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, 1988, pp. 314–329.

[2] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 850–857, June 1995.

[3] T. V. 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, June 1997.

[4] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," Tech. Rep. DCS-TR-314, Rutgers Univ., New Brunswick, NJ, Oct. 1994.

[5] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, no. 4, Dec. 1995.

[6] H. Balakrishnan, N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Networking*, vol. 5, pp. 756–769, Dec. 1997.

[7] V. Jacobson. (1990.) "Berkeley TCP evolution from 4.3-Tahoe to 4.3-Reno," in *Proc. 18th Internet Engineering Task Force* [Online]. Available HTTP: http://ietf.org/proceedings/directory.html

[8] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated, Vol. II: The Implementation*. Reading, MA: Addison-Wesley, 1995.

[9] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Comput. Commun. Rev.*, vol. 21, no. 2, pp. 26–42, Apr. 1991.

[10] S. Shenker, L. Zhang, and D. Clark, "Some observations on the dynamics of a congestion control algorithm," *Comput. Commun. Rev.*, vol. 20, no. 5, pp. 30–39, Oct. 1990.

[11] L. Zhang, S. Shenker, and D. Clark, "Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic," in *Proc. ACM SIGCOMM*, 1991, pp. 133–147.

[12] D. E. Comer, *Internetworking with TCP/IP, Vol. I: Principles, Protocols, and Architecture*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[13] B. Suter, T. V. Lakshman, D. Stiliadis, and A. Choudhury, "Design considerations for supporting TCP with per-flow queuing," in *Proc. IEEE INFOCOM*, 1998, pp. 299–306.

[14] R. Guérin, S. Kamat, V. Peris, and R. Rajan. (1998.) "Scalable QoS provision through buffer management," in *Proc. ACM SIGCOMM* [Online]. Available HTTP: http://ietf.org/proceedings/directory.html

[15] 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," in *Proc. IEEE INFOCOM*, 1997, pp. 1199–1209.

[16] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, June 1993.

[17] H. Chaskar and U. Madhow, "Fair scheduling with tunable latency: A Round Robin approach," in *Proc. IEEE GLOBECOM*, 1999, to be published.

[18] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: IEEE Press, 1993.

[19] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Baltimore, MD: The Johns Hopkins Press, 1981.

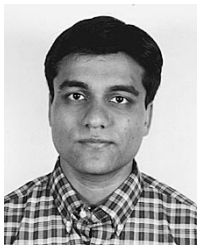
[20] G. L. Stüber, *Principles of Mobile Communication*. Norwell, MA: Kluwer, 1996.

[21] H. S. Wang and N. Moayeri, "Finite state Markov channel—A useful model for radio communications," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 163–171, Feb. 1995.

[22] H. S. Wang and P.-C. Chang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 353–357, May 1996.

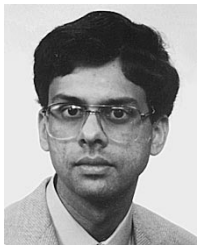
[23] W. Turin and R. van Nobelen, "Hidden Markov modeling of flat fading channels," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1809–1817, Dec. 1998.

[24] C. S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 913–931, May 1994.



Hemant M. Chaskar (M'99) was born in Pune, India, on January 26, 1972. He received the B.Eng. degree from the University of Pune, India, in 1993, and the M.Eng. degree from the Indian Institute of Science, Bangalore, in 1995, both in electronics and telecommunication engineering. He received Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1999.

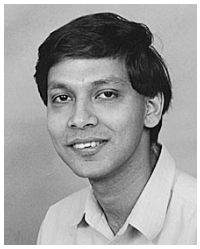
From 1995 to 1999, he was a Research Assistant in the Coordinated Sciences Laboratory, University of Illinois. He spent the summer of 1997 in the Performance Analysis Department of Lucent Technologies-Bell Labs, Holmdel, NJ. Currently, he is a Research Engineer in the Internet Multimedia Networks Department at Nokia Research Center, Burlington, MA. His research interests are in wireless communications, high-speed networks, communication theory, and applied mathematics.



T. V. Lakshman (S'84–M'85–SM'98) received the the Master's degree in physics from the Indian Institute of Science, Bangalore, India., and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1986.

From 1986 to 1995, he was at Bellcore where he was most recently a Senior Research Scientist and Technical Project Manager in the Information Networking Research Laboratory. He is currently a Distinguished Member of Technical Staff in the High-Speed Networks Research Department at Bell Labs, Holmdel, NJ. His recent research has included issues related to traffic characterization and provision of quality of service, architectures and algorithms for gigabit IP routers, end-to-end flow control in high-speed networks, traffic shaping and policing, switch scheduling, and routing in MPLS networks. His current research interests are in the areas of high-speed networking, distributed computing, and multimedia systems.

Dr. Lakshman is an Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING. He is a co-recipient of the 1995 ACM Sigmetrics/Performance Conference Outstanding Paper Award, and the IEEE Communications Society 1999 Fred. W. Ellersick Prize Paper Award.



U. Madhow (S'86–M'90–SM'96) received the B.Eng. degree from the Indian Institute of Technology, Kanpur, in 1985. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana-Champaign, in 1987 and 1990, respectively.

From 1990 to 1991, he was a Visiting Assistant Professor at the University of Illinois at Urbana-Champaign. From 1991 to 1994, he was a Research Scientist at Bell Communications Research, Morristown, NJ. Since 1994, he has been with the Department of Electrical and Computer Engineering at the University of Illinois, where he is currently an Associate Professor. His current research interests are in communication systems and networking, with current emphasis in wireless communications and high-speed networks.

Dr. Madhow is an Associate Editor for Spread Spectrum for the IEEE TRANSACTIONS ON COMMUNICATIONS, and an Associate Editor for Detection and Estimation for the IEEE TRANSACTIONS ON INFORMATION THEORY. He is a recipient of the NSF CAREER Award.