Transfer Delay Analysis of WAP 2.0 for Short-Lived Flows

Humphrey Rutagemwa, *Student Member, IEEE,* Xuemin (Sherman) Shen, *Senior Member, IEEE,* and Jon W. Mark, *Life Fellow, IEEE*

***Abstract*—In this paper, an analytical framework for studying the transfer delay of wireless application protocol (WAP) 2.0 for short-lived flows is developed, which is based on a two-state Markov chain that approximates both correlated and independent packet losses. For a given wireless link and protocol parameters, an explicit mathematical expression which yields a good estimate of the WAP 2.0 transfer delay is derived. The analytical results are validated by simulation. It is shown that for large file sizes ($>$ 30 kB), WAP 2.0 is more sensitive to bursty packet losses than random packet losses. It is also shown that the transfer delay of WAP 2.0 can be improved by increasing the size of the initial window in a low-rate bursty error environment but degrades in a high-rate bursty error environment.**

***Index Terms*—Markov model, performance analysis, short-lived flows, wireless application protocol (WAP) 2.0, wireless profiled (WP)-transmission control protocol (TCP).**

I. Introduction

WIRELESS application protocol (WAP) has been attracting many attention recently, as it provides a standard environment for wireless mobile communications. WAP is proposed by the WAP Forum [1], [2] (now managed by Open Mobile Alliance) to enable mobile users with digital handheld devices to access the Internet and advanced telephony services. It is a *de facto* world standard for the presentation and delivery of wireless information services on wireless devices. The earlier versions, i.e., WAP 1.x, were standards that are optimized for mobile environment, where handheld wireless devices are limited by central processing unit power, memory, battery lifetime, and simple user interface, and wireless links are characterized by low bandwidth, high latency, and unpredictable availability and stability. As wireless networks and mobile devices evolve, some of these constraints become less significant. The WAP 2.0 standard, which was released in July 2001, utilizes the advantages of advancement in wireless network and mobile devices.

Similar to the Open Systems Interconnection reference model, WAP 2.0 has a layered architecture, with an application framework and a protocol framework. The application framework is built on top of the protocol framework and provides an interoperable environment. It allows the applications and services to be built on a wide variety of different wireless platforms. The protocol framework has four layers, namely 1) a session service layer, 2) a transfer service layer, 3) a transport service layer, and 4) a bearer networking layer. The transport service layer provides datagram service and connection services. User datagram protocol (UDP) and wireless datagram protocol (WDP) are used to provide datagram transport service, and transmission control protocol (TCP) is used to provide connection-oriented transport service. Since standard TCP tends to perform poorly in a wireless network, wireless profiled-TCP (WP-TCP), which is fully compatible with TCP, has been adopted to cope with the wireless network characteristics. A complete overview of the WAP 2.0 architecture is given in [1].

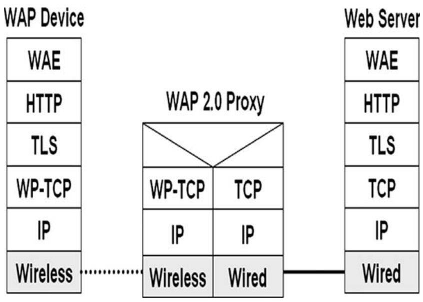
WP-TCP uses a window-based congestion control mechanism [3], [4]. The WP-TCP sender maintains a congestion window ($cwnd$), which limits the number of outstanding unacknowledged data segments in the network, and a slow start threshold ($ssthresh$), which determines the rate of adjusting the $cwnd$. On startup, $cwnd$ and $ssthresh$ are initialized to *initial window* and *maximum window size*, respectively. Whenever a new acknowledgement (ACK) is received, the $cwnd$ is increased by one segment if it is below $ssthresh$ (slow start phase) and by $1/cwnd$ if it is equal to or greater than $ssthresh$ (congestion avoidance phase). In either phase, the upper limit of increasing $cwnd$ is *maximum window size*. The WP-TCP sender assumes a packet is lost either after a timeout or after receiving a certain number of consecutive *duplicate* ACKs (ACK with the sequence number same as the previous ACK). This number is normally referred to as the *duplicate* ACK threshold. When a timeout occurs, the $ssthresh$ is set to $\max\{2, cwnd/2\}$, and the $cwnd$ is reset to 1. The lowest unacknowledged packet is retransmitted and the WP-TCP sender enters the slow start phase. In the case of packet loss, which is indicated by *duplicate* ACKs, fast retransmit is invoked followed by fast recovery. The fast recovery procedure ensures that the congestion avoidance phase follows after fast retransmit and not the slow start phase. Details on WP-TCP specification can be found in [5].

Various studies have shown that short-lived flows dominate most of the Internet traffic, e.g., [6] and [7]. In order for WAP to continue providing solutions for connection-oriented transport service in wireless mobile communications, the performance of short-lived WAP flows over wireless links needs to be thoroughly studied. There are several approaches that can be used to study the performance behavior of WAP. One of these is mathematical modeling. This approach is widely accepted,

Manuscript received May 27, 2004; revised February 7, 2005 and March 9, 2005. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Strategic Grant STPGP 257682. The review of this paper was coordinated by Prof. T. Hou.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: humphrey@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca; jwmark@bbcr.uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2007.895469



AQ1 Fig. 1. Block diagram of a WAP 2.0 wireless/Internet interworking model.

as it is fast, flexible, and cost-effective while producing results with reasonable accuracy. Research on WAP performance based on WAP 1.x standards has been conducted since its appearance [8]–[10]. To the best of our knowledge, no research on performance modeling for the WAP 2.0 standard has been reported in the literature. Since WAP 2.0 uses WP-TCP to provide connection-oriented transport service, previous models of standard TCP for short-lived flows may be valid. Several models have been proposed for short TCP transfers [11]–[14]; however, they are based on an independent packet loss assumption, which does not fit well in the wireless environment. In this paper, an analytical framework for studying the transfer delay of short-lived WAP 2.0 flows is developed, which is based on a two-state Markov chain that approximates both correlated and independent packet losses. An explicit mathematical expression is derived based on the channel model, which represents a reasonable estimate of the approximate file transfer time, for given wireless link and protocol parameters. Simulation results are provided to validate the proposed analytical approach. The rest of this paper is organized as follows: The system model is developed in Section II, and the analysis of WAP transfer delay is given in Section III. Section IV presents the simulation results, and the concluding remarks are given in Section V.

II. System Model

Throughout this paper, the following notations will be used:

$\lceil x \rceil$	Rounds x to the nearest integer greater than or equal to x.
$\lfloor x \rfloor$	Rounds x to the nearest integer smaller than or equal to x.
BL	Burst length.
e	Average error rate.
l	Channel delay.
$cwnd$	Congestion window.
$ssthresh$	Slow start threshold.
W_{iw}	Initial window.
$W_{\max}$	Maximum window size.
d_{AB}	Transition time from state A to state B.
ϕ_{AB}	Transition probability from state A to state B.
ρ	Packet length measured in bytes.
F	File size measured in bytes.
$N = \lceil F/\rho \rceil$	File size measured in packets.

Fig. 1 shows the WAP 2.0 wireless/Internet internetworking model with three types of nodes, namely 1) fixed host (FH), 2) base stations (BS), and 3) mobile host (MH). Standard TCP is used for communication between FH (Web Server) and BS (WAP 2.0 proxy), whereas WP-TCP is used for communication between BS and MH (WAP Client). In this model, WAP 2.0 utilizes proxy technology to connect the wireless domain and the Internet. Since the main interest in this paper is to study the impacts of the wireless links on the transfer delay of WAP 2.0 for short-lived flows, we focus attention on the wireless domain only.

To analyze the impact of the wireless link on the WAP 2.0 transfer delay, the WAP 2.0 architecture is profiled into three layers, namely 1) application, 2) transport, and 3) network. The application layer generates a predefined type of data traffic. In this paper, a small file is considered to be transferred by using file transfer protocol (FTP)/Web like application. The transport layer reliably delivers the generated data packets over a single link wireless channel defined in the network layer. WP-TCP is considered as the transport layer protocol and implemented with all mandatory requirements (RFC 0793, RFC 1122 [15], and RFC 2581 [3]) and some important optional requirements (large initial window RFC 2414 [16], selective acknowledgement (SACK) RFC 2018 [17], and timestamps option RFC 1323 for roundtrip-time measurement). Note that the SACK option enables the WP-TCP receiver to report blocks of packet losses to the WP-TCP sender, whereas the timestamps option enables the WP-TCP sender to estimate roundtrip time fairly accurately. For the wireless channel, a nonline-of-sight frequency-nonselective (flat) multipath fading channel with packet transmission rate (in packets/second) that is much higher than the maximum Doppler frequency (in hertz) is assumed. By considering a modulation scheme, the dynamics of the fading channel can be characterized at the packet level. However, the performance analysis of high-level protocols becomes quite complex. As an alternative to this problem, a widely adopted two-state Markov channel model [18], [19] is used to approximate the error process at the packet level. The two-state Markov channel model has a good (g) state and a bad (b) state. Packet loss probability is 1 in the bad state and 0 in the good state. The transition probabilities of these states are given by the matrix

$$\mathbf{P} = \begin{pmatrix} p_{\mathrm{bb}} & p_{\mathrm{bg}} \\ p_{\mathrm{gb}} & p_{\mathrm{gg}} \end{pmatrix} \tag{1}$$

where p_{xy} is the transition probability from channel state $x \in \{\mathrm{g}, \mathrm{b}\}$ to channel state $y \in \{\mathrm{g}, \mathrm{b}\}$. Given an average error rate e, which is defined as the ratio of the number of erroneous packets to the total number of packets sent, and a burst length BL, which is defined as the average length of consecutive packet losses, the transition probabilities can be computed as

$$p_{\mathrm{bg}} = \frac{1}{BL} \tag{2}$$

and

$$p_{\mathrm{gb}} = \frac{1}{BL}\left(\frac{e}{1-e}\right). \tag{3}$$

Note that $p_{\mathrm{bb}} = 1 - p_{\mathrm{bg}}$ and that $p_{\mathrm{gg}} = 1 - p_{\mathrm{gb}}$. The derivation of the fading parameters BL and e from a low-level channel error process is beyond the scope of this paper, but a detailed discussion can be found in [18].

III. Performance Analysis

A single-pair WP-TCP sender–receiver is assumed to run over the wireless channel whose packet error process is modeled by the two-state Markov model. Since packet transmission time is assumed to be shorter than the channel coherence time, it is reasonable to consider the state transitions of the Markov channel model after every time slot (the time to transmit one packet). The metric of our interest is file transfer time (transfer delay), which is defined as the time used to download a given file size. WP-TCP has three transfer stages, namely 1) connection establishment, 2) data transfer, and 3) connection tearing down. Since we are only interested in the completion time of transferring the actual data packets, the connection tearing down stage is not considered. In case there is a need to consider this stage, the modeling process of the connection tearing-down stage is similar to that for the connection establishment stage. Therefore, the average file transfer time T can be found as follows: $T = T_{\text{cs}} + T_{\text{dt}}$, where T_{cs} is the average connection setup time, and T_{dt} is the average data transfer time.

A. Connection Establishment

The connection establishment stage is a "three-way handshake," as described in [20]. In this stage, the occurrence of packet losses is considered in both the downlink direction (from BS to MH) and the uplink direction (from MH to BS). Since the SYN[1] timeout is relatively large and doubles for every transmission retry, packet losses are considered to occur independently. By assuming the same SYN timeout T_s and average error rate $(e < 0.5)$ in the downlink and uplink directions and by considering an infinite number of retries for connection establishment, the average connection setup time can be calculated as [11]

$$T_{\text{cs}} = 2l + T_s \left(\frac{2e}{1 - 2e} \right) \tag{4}$$

where l is the channel delay that is defined as the time taken for a packet traveling from the BS to the MH or vice-versa. It is a result of propagation delay and all processing delays encountered at the end nodes. Note that the time to transmit SYN or SYN-ACK is assumed to be negligible, and thus, the roundtrip time is approximated by $2l$. This assumption is reasonable since the size of SYN and SYN-ACK packets is very small compared with the size of data packets.

B. Data Transfer

The data transfer stage begins right after the connection establishment stage and ends when the sender receives an acknowledgement for the last byte of the transferred data. Fig. 2 shows the approximate model for the actual WP-TCP data transfer stage. The sender begins transmission in the slow start phase until either data transfer is completed (transition a), the channel changes to the bad state (transition b), or the $cwnd$ reaches the $ssthresh$ (transition c). Similarly, the sender remains in the congestion avoidance phase until either data transfer is completed (transition d) or the channel changes to the bad state (transition e). In the loss recovery phase, the WP-TCP sender uses duplicate acknowledgement or timeout mechanisms to detect and recover lost packets. If timeout is used to detect packet loss, the sender enters the slow start phase (transition f) and restarts retransmission. If the duplicate acknowledgment mechanism is used to detect and recover lost packet(s), the sender enters the congestion avoidance state (transition g) and continues with transmission or finishes if the recovered packet completes the transmission (transition h).

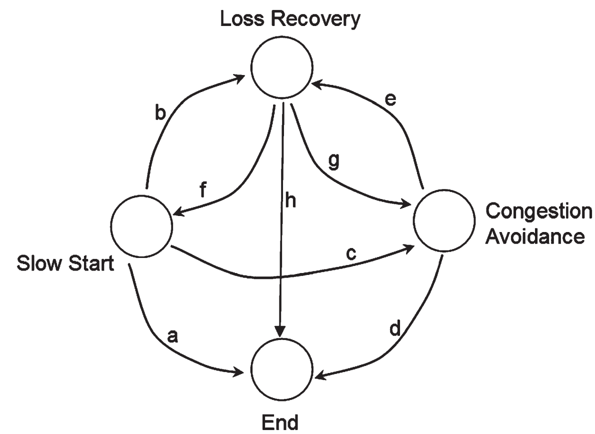


Fig. 2. Approximate model for data transfer stage.

To track the data transfer process, we consider a random process $S(t) = (c(t-1), w(t), w_{\text{th}}(t), k(t))$, where $c(t) \in \{\text{b}, \text{g}\}$ is the channel state, $w(t) \in \{1, 2, \ldots, W_{\max}\}$ is the $cwnd$, $w_{\text{th}}(t) \in \{2, 3, \ldots, \lceil W_{\max}/2 \rceil, W_{\max}\}$ is the $ssthresh$, $k(t) \in \{0, 1, \ldots, N\}$ is the number of packets transmitted successfully, t is the time measured in slots,[2] $W_{\max}$ is the maximum window size, and N is the file size measured in packets. Given a packet length ρ in bytes and the size of the transfer file F in bytes, N can be computed as $N = \lceil F/\rho \rceil$. Note that according to WP-TCP congestion control, the $w_{\text{th}}(t)$ is initialized to $W_{\max}$ on startup and can be reset to $\max\{2, \lceil W_{\max}/2 \rceil\}$ whenever required. Since $w(t)$ cannot exceed $W_{\max}$, the possible values for $w_{\text{th}}(t)$ are $2, 3, \ldots, \lceil W_{\max}/2 \rceil$, and $W_{\max}$. By sampling the random process $S(t)$ at instants t_n just after transition occurs, the new sampled process $S(t_n)$ is a semi-Markov process [21] with embedded Markov chain, in the state space $W_S = \{(c, w, w_{\text{th}}, k) : c \in \{\{\text{b}, \text{g}\}, w \in \{1, 2, \ldots, W_{\max}\}, w_{\text{th}} \in \{2, 3, \ldots, \lceil W_{\max}/2 \rceil, W_{\max}\}, k \in \{0, 1, \ldots, N\}\}$, which is defined by the transition probability matrix $\mathbf{\Phi} = [\phi_{AB}]$. ϕ_{AB} denotes the transition probability from state $A \in W_S$ to state $B \in W_S$. To compute the average data transfer time, the transitions of embedded Markov chain $S(t_n)$ are labeled with corresponding elements of the matrix $\mathbf{D} = [d_{AB}]$, where element d_{AB} is the transition time from state $A \in W_S$ to state $B \in W_S$. To reduce the state space and computational complexity, all possible states with

[1]SYN is the initial packet sent by the WP-TCP sender to establish communication.

[2]A slot is equal to the time used to transmit one WP-TCP data packet.

which the data transfer stage can end are grouped to form a super state, which is denoted as (X, X, X, N). Since all states forming the super state (X, X, X, N) are trapping states (i.e., they have no self-transitions), the vector $\mathbf{T} = [T_A]$, whose elements represent the expected delay, given that the transmission process starts in the initial state $A \in W_S$ and terminates at the super state (X, X, X, N), can be computed as [21]

$$\mathbf{T} = [\mathbf{I} - \mathbf{\Phi}'']^{-1}\overline{\mathbf{D}} \tag{5}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{\Phi}''$ is obtained by deleting a row and a column corresponding to the super state (X, X, X, N) from $\mathbf{\Phi}$, and $\overline{\mathbf{D}} = \mathrm{diag}(\mathbf{\Phi}'[\mathbf{D}']^t)$, where the superscript t denotes matrix transpose. $\mathbf{\Phi}'$ and $\mathbf{D}'$ are obtained by only deleting a row corresponding to the super state (X, X, X, N) from $\mathbf{\Phi}$ and $\mathbf{D}$, respectively. By considering the WP-TCP initialization procedures introduced in Section I, the possible data transfer initial states are $(\mathrm{g}, W_{\mathrm{iw}}, W_{\max}, 0)$ and $(\mathrm{b}, W_{\mathrm{iw}}, W_{\max}, 0)$. Consequently, the average data transfer time can be computed as

$$T_{\mathrm{dt}} = \pi_{\mathrm{g}} T_{(\mathrm{g}, W_{\mathrm{iw}}, W_{\max}, 0)} + \pi_{\mathrm{b}} T_{(\mathrm{b}, W_{\mathrm{iw}}, W_{\max}, 0)} \tag{6}$$

where π_{g} and π_{b} are the steady-state probabilities of the channel being in states g and b, respectively. π_{g} and π_{b} can be computed from the transition probability matrix $\mathbf{P}$, as shown in [22]. To compute the elements of the matrices $\mathbf{\Phi}$ and $\mathbf{D}$, the transmission process in the data transfer stage is analyzed, as depicted in Fig. 2, in three phases, namely 1) slow start, 2) congestion avoidance, and 3) loss recovery. In each phase, the initial states and possible transitions to the final states are explored by conditioning the number of packets transmitted successfully. Then, for every possible transition, the corresponding transition probability and transition time are computed. Note that this model is not expected to produce exact value for data transfer time but rather a reasonable approximation.

1) Slow Start: The transmission process enters the slow start phase either at the beginning of data transfer or when a packet loss is recovered by using the timeout mechanism. Therefore, a set of initial states is $A = \{(\mathrm{g}, W_{\mathrm{iw}}, W_{\max}, 0) \cup [(\mathrm{g}, 1, w_{\mathrm{th}}, k) : w_{\mathrm{th}} \in \{2, 3, \ldots, \lceil\{W_{\max}/2\rceil\}, k \in \{1, 2, \ldots, N-1\}]\}$, where $\cup$ is the union operation. Let $s \in \{1, 2, \ldots, N-k\}$ be the number of packets transmitted successfully before a transition occurs (transfer ends or enters congestion avoidance or packet loss). Under perfect channel condition, the latency and dynamic of congestion window in the slow start phase can accurately be modeled [20]. Given the initial state $(\mathrm{g}, w, w_{\mathrm{th}}, k)$, the possible final states can be deterministically computed as

$$B = \begin{cases} (X, X, X, N), & s \le 2w_{\mathrm{th}} - w, \quad s = N-k \\ \left(\mathrm{b}, W_B^{\mathrm{SS}}(s), w_{\mathrm{th}}, k+s\right), & s < 2w_{\mathrm{th}} - w, \quad s < N-k \\ \left(\mathrm{g}, W_B^{\mathrm{SS}}(s), w_{\mathrm{th}}, k+s\right), & s = 2w_{\mathrm{th}} - w, \quad s < N-k \end{cases} \tag{7}$$

where $W_B^{\mathrm{SS}}(s) = \lfloor (w+s)/2 \rfloor$ is the $cwnd$ after sending s packets. Note that the first case, second case, and third case in (7) correspond to transitions a, b, and c presented in Fig. 2, respectively. The associated transition time can be obtained by

$$d_{AB} = \begin{cases} s, & Q < 1 \\ s + (2l+1)Q - (2^Q - 1)w, & \text{otherwise} \end{cases} \tag{8}$$

where $Q = \lfloor \min\{1 + \log_2(2l/w + 1/w), \log_2(1 + s/w - 1/w)\} \rfloor$ and l is the channel delay measured in slots. The transition probability can be found as

$$\phi_{AB} = \begin{cases} p_{\mathrm{gg}}^{s-1}, & s = \min\{N-k, 2w_{\mathrm{th}} - w\} \\ p_{\mathrm{gg}}^{s-1} p_{\mathrm{gb}}, & s < \min\{N-k, 2w_{\mathrm{th}} - w\} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

2) Congestion Avoidance: Since the transmission process enters the congestion avoidance phase when a packet loss is recovered by using the duplicate acknowledgment mechanism or when the $cwnd$ reaches the $ssthresh$, a set of initial states is therefore given as $A = \{(\mathrm{g}, w_{\mathrm{th}}, w_{\mathrm{th}}, k) : w_{\mathrm{th}} \in \{2, 3, \ldots, \lceil W_{\max}/2 \rceil, W_{\max}\}, k \in \{1, 2, \ldots, N-1\}\}$. Let $s \in \{1, 2, \ldots, N-k\}$ be the number of packets transmitted successfully before transition occurs (transfer ends or packet loss occurs). Under perfect channel condition, the latency and dynamic of congestion window in the congestion avoidance phase can be modeled similar to that in [20]. Given the initial state $(\mathrm{g}, w_{\mathrm{th}}, w_{\mathrm{th}}, k)$, the possible final states can be deterministically obtained as

$$B = \begin{cases} (X, X, X, N), & s = N-k \\ \left(\mathrm{b}, W_A^{\mathrm{CA}}(s), w_{\mathrm{th}}, k+s\right), & s < N-k \end{cases} \tag{10}$$

where $W_B^{\mathrm{CA}}(s) = \min\{-1/2 + \sqrt{(w_{\mathrm{th}} - 1/2)^2 + 2s}, W_{\max}\}$. Note that the first case and second case in (10) correspond to transitions d and e presented in Fig. 2, respectively. The associated transition time can be found as

$$d_{AB} = s + (2l + 2 - w_{\mathrm{th}})Q - (Q+1)Q/2 \tag{11}$$

where $Q = \min\{2l - w_{\mathrm{th}} + 2, 1/2 - w_{\mathrm{th}} + \sqrt{(w_{\mathrm{th}} - 1/2)^2 + 2s}\}$. The transition probability can be written as

$$\phi_{AB} = \begin{cases} p_{\mathrm{gg}}^{s-1}, & s = N-k \\ p_{\mathrm{gg}}^{s-1} p_{\mathrm{gb}}, & s < N-k \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

3) Loss Recovery: All transitions to the loss recovery phase occur when the channel becomes bad during transmission and when the first packet loss occurs in the first time slot of the loss recovery phase. Therefore, a set of initial states is given as $A = \{(\mathrm{b}, w, w_{\mathrm{th}}, k) : w \in \{1, 2, \ldots, W_{\max}\}, w_{\mathrm{th}} \in \{2, 3, \ldots, \lceil W_{\max}/2 \rceil, W_{\max}\}, k \in \{0, 1, \ldots, N-1\}\}$. After the first packet loss, all subsequent packets that are transmitted successfully will generate *duplicate* ACKs. Let Z be the *duplicate* ACK *threshold*. If the total number of *duplicate* ACKs is less than Z, the sender will wait for a timeout interval T_0 and then enter the slow start phase. When the number of *duplicate* ACKs reaches Z, fast transmit followed by fast recovery will be triggered. During fast recovery, *duplicate* ACKs are ignored until half of the window is acknowledged, and after that, the WP-TCP sender will send a new segment for every received duplicate ACK. To model the recovery process, two cases are considered, namely 1) when a lost packet is recovered with the timeout mechanism and 2) when it is recovered without using the timeout mechanism.

Let $\alpha_c(s,w)$ be defined as the probability of having $s \in \{1, 2, \ldots, w-1\}$ successfully transmitted packets out of w transmitted packets, given that the channel was in the bad state at the beginning of transmission and that the channel is in state $c \in \{\mathrm{g}, \mathrm{b}\}$ at the end of transmission. Similar to [23], it follows that

$$\alpha_{\mathrm{b}}(s,w) = \begin{cases} p_{\mathrm{bb}}^{w}, & s=0 \\ \sum_{i=1}^{\min\{s,w-s\}} \binom{w-s}{i}\binom{s-1}{i-1} p_{\mathrm{bb}}^{w-s-i} \\ \qquad \cdot\, p_{\mathrm{bg}}^{i} p_{\mathrm{gb}}^{i} p_{\mathrm{gg}}^{s-i}, & s=1,\ldots,w-1 \\ 0, & s \geq w \end{cases} \tag{13}$$

and

$$\alpha_{\mathrm{g}}(s,w) = \begin{cases} \sum_{i=1}^{\min\{s+1,w-s\}} \binom{w-s-1}{i-1}\binom{s}{i-1} p_{\mathrm{bb}}^{w-s-i} \\ \qquad \cdot\, p_{\mathrm{bg}}^{i} p_{\mathrm{gb}}^{i-1} p_{\mathrm{gg}}^{s-i+1}, & s=0,\ldots,w-1 \\ 0, & s \geq w \end{cases} \tag{14}$$

Case I—Loss Recovery With Timeout Mechanism: We further consider two scenarios that will result in timeout. The first scenario is when s is less than Z. The second scenario is when s is greater than or equal to Z but less than half of the $cwnd$ and packets that are retransmitted using fast transmit get lost. From the empirical studies on TCP Reno [24], it is found that most of the TCP flows only suffer a single timeout. With this observation, a timeout with no exponential backoff is assumed. From Section II, WP-TCP is assumed to be implemented with the timestamps option for roundtrip-time measurement. This option enables the WP-TCP sender to estimate the roundtrip time fairly accurately. Consequently, in this analysis, T_0 is set to twice the roundtrip time [i.e., $T_0 = 2(2l+1)$]. Given the initial state $(\mathrm{b}, w, w_{\mathrm{th}}, k)$, possible final states can be written as

$$B = (c, 1, \lceil w/2 \rceil, k+s) \tag{15}$$

for $c \in \{\mathrm{g}, \mathrm{b}\}$. The transition time and transition probability can be written as

$$d_{AB} = \begin{cases} T_0, & s < Z \\ T_0, & Z \leq s, \quad s < w/2 \end{cases} \tag{16}$$

and ϕ_{AB} is defined in (17), shown at the bottom of the page, where $p_{xy}(n)$ is the n-step transition probability from channel state $x \in \{\mathrm{b}, \mathrm{g}\}$ to channel state $y \in \{\mathrm{b}, \mathrm{g}\}$.

Case II—Loss Recovery Without Timeout Mechanism: When the loss recovery process is completed without using the timeout mechanism, the sender continues with transmission of new packets if there are more data to send (i.e., if $w+k<N$); otherwise, transmission ends (i.e., if $w+k=N$). Given the initial state $(\mathrm{b}, w, w_{\mathrm{th}}, k)$, possible final states can be written as

$$B = \begin{cases} (X, X, X, N), & w+k=N \\ (c, w/2, \lceil w/2 \rceil, w+k), & w+k<N \end{cases} \tag{18}$$

for $c \in \{\mathrm{g}, \mathrm{b}\}$. During the loss recovery phase, it is possible for retransmitted packets to get lost again. However, if the SACK option is enabled and the number of successfully transmitted packets is larger than Z, the impact of retransmitted packet losses becomes less significant. Since, in this case, the number of successfully transmitted packets is greater than Z, perfect retransmissions are assumed (i.e., lost packets are always retransmitted successfully).

To compute the transition time and transition probability, two scenarios are further considered. The first scenario is when s is greater than or equal to Z and also greater than or equal to half of the $cwnd$. With the perfect retransmission assumptions, the number of packets recovered per roundtrip will equal the number of *duplicate* ACKs exceeding the $cwnd$ (i.e., $s - w/2$). Therefore, the time spent in recovering lost packets can be approximated as $(1+2l)\lceil (w-s)/(1+(s-w/2)) \rceil$, where $(w-s)$ is the number of lost packets, and $(1+2l)$ is the roundtrip time measured in time slots. The second scenario is when s is greater than or equal to Z but less than half of the congestion window. Again, with perfect retransmission assumptions, only one packet can be recovered per roundtrip. Therefore, the time spent in recovering lost packets can be approximated as $(1+2l)(w-s)$. Let $v(s,w)$ denote the time spent to complete the loss recovery phase, given that at the beginning of the recovery phase, s packets out of w transmitted packets were successful. We have the following:

$$v(s,w) = \begin{cases} (w+2l) + (1+2l)\left\lceil \frac{(w-s)}{1+(s-w/2)} \right\rceil, & \frac{w}{2} \leq s \\ (w+2l) + (1+2l)(w-s), & s < \frac{w}{2}. \end{cases} \tag{19}$$

Let $\beta(s,w,k)$ denote the probability of completing the loss recovery phase without using a timeout, given that s packets out of w packets were initially transmitted successfully. Then, the probability $\beta(s,w,k)$, which is defined in (20), can be written as shown at the bottom of the next page.

From (19) and (20), the transition time and transition probability can be computed by considering s over the interval $[Z, w-1]$ as

$$d_{AB} = \frac{\sum_{s=Z}^{w-1} v(s,w)\beta(s,w,k)}{\sum_{s=Z}^{w-1} \beta(s,w,k)} \tag{21}$$

$$\phi_{AB} = \begin{cases} \left[\alpha_{\mathrm{g}}(s,w) + \alpha_{\mathrm{b}}(s,w)\right] p_{\mathrm{bc}}(T_0 - 1), & s < Z \\ \left[\alpha_{\mathrm{g}}(s,w) + \alpha_{\mathrm{b}}(s,w)\right] p_{\mathrm{bb}}(w - s + Z + 2l) \cdot p_{\mathrm{bc}}(T_0 - 1), & Z \leq s, \quad s < \frac{w}{2} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

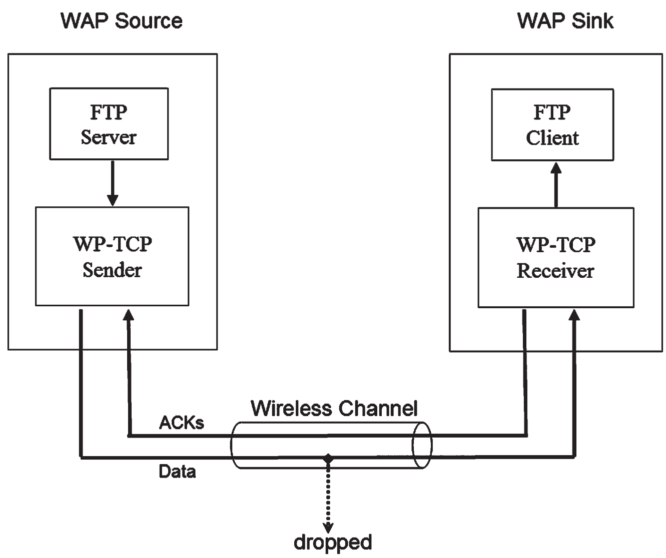


Fig. 3. Simulation model.

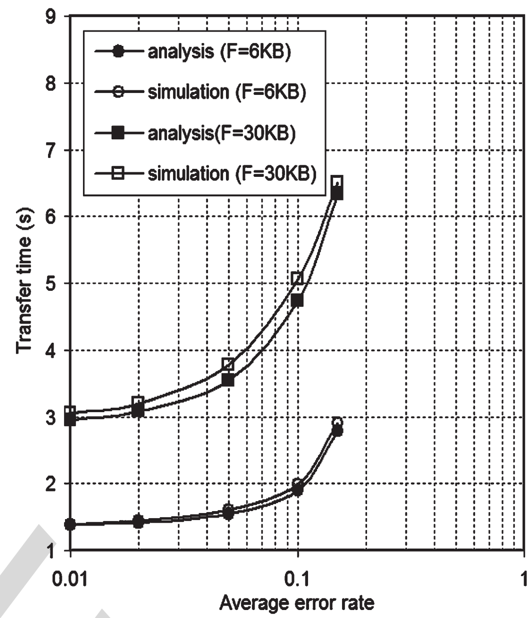


Fig. 4. Transfer time versus e for $BL = 1$, $F = 6$, and 30 kB.

and

$$\phi_{AB} = \sum_{s=Z}^{w-1} \beta(s, w, k). \tag{22}$$

IV. Simulation Results

In this section, the proposed analytical model is validated by simulations using the ns-2 simulator [25]. Fig. 3 shows the simulation model. A single pair of WP-TCP sender–receiver is configured to run over the wireless link with the packet error process modeled by a two-state Markov chain. FTP is set as an application that transfers a file with a specified size.

To obtain more accurate results, each simulation scenario is repeated 100 times with different random seeds to arrive at the average results. The analytical and simulation results are obtained by setting the bandwidth B at 128 kb/s, the *duplicate* ACK *threshold* Z at 3, and the packet length ρ at 1 kB, while the channel delay l, the maximum window size $W_{\max}$, the burst length BL, the file size F, the average error rate e, and the initial window W_{iw} take on different values.

A. Effect of the Average Error Rate and the Burst Length

BL reflects the correlation of the packet errors. If the BL is relatively large, packet errors are highly correlated (burst errors). On the other hand, if the BL is relatively small, the packet errors occur independently (random errors). In this paper, the transmission is considered to be under a random error environment if $BL = 1$ and bursty error environment if $BL \geq 2$.

For $W_{\mathrm{iw}} = 1$, $l = 125$ ms, $W_{\max} = 8$, and $F = 6$ and 30 kB, the impact of average error rate in the random error environment ($BL = 1$) and bursty error environment ($BL = 3$) are shown in Figs. 4 and 5, respectively. It can be seen that the transfer time increases with average error rate. This is because of the burden added due to packet retransmission. For small file size (6 kB), the impact of average error rate in the bursty error and random error environments appears to be almost the same. However, for the large file size (30 kB), the impact of error rate is more significant in the bursty error environment (Fig. 5) than in the random error environment (Fig. 4). For instance, when the average error rate is 0.01, the difference in transfer time when $BL = 1$ and $BL = 3$ is nearly half a second. However, when the average error rate increases to 0.15, the transfer time for $BL = 3$ is 2 s more than that for $BL = 1$. The reason for these observations can be explained as follows. In the case of small files, the *cwnd* tends to be small, and therefore, any packet loss will most likely cause a timeout to trigger. Hence, bursty losses and random losses exert almost the same effect to transfer time. In the case of large files, *cwnd* tends to grow relatively large such that packet losses can be detected by duplicate acknowledgements. Since, in the bursty error environment, most of the corrupted packets are from the same sender window, the advantage of having

$$\beta(s,w,k) = \begin{cases} [\alpha_{\mathrm{g}}(s,w)+\alpha_{\mathrm{b}}(s,w)] \cdot p_{\mathrm{bc}}\left(v(s,w)\right), & \frac{w}{2} \leq s, \quad w+k<N \\ \alpha_{\mathrm{g}}(s,w)+\alpha_{\mathrm{b}}(s,w), & \frac{w}{2} \leq s, \quad w+k=N \\ [\alpha_{\mathrm{g}}(s,w)+\alpha_{\mathrm{b}}(s,w)]\, p_{\mathrm{bg}}(w-s+Z+2l) \cdot p_{\mathrm{bc}}\left(v(s,w)\right), & s<\frac{w}{2}, \quad w+k<N \\ [\alpha_{\mathrm{g}}(s,w)+\alpha_{\mathrm{b}}(s,w)] \cdot p_{\mathrm{bg}}(w-s+Z+2l), & s<\frac{w}{2}, \quad w+k=N \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

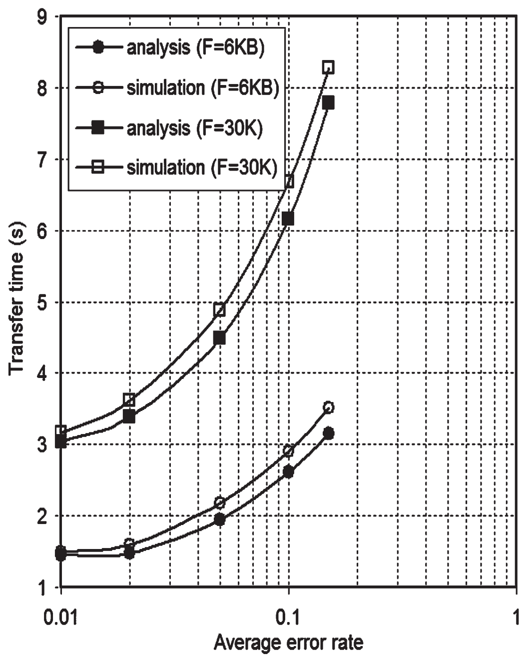


Fig. 5. Transfer time versus e for $BL = 3$, $F = 6$, and 30 kB.

TABLE I
TRANSFER TIMES (IN SECONDS) FOR $e = 0.01$

F (KB)	W_{iw} = 1		W_{iw} = 4	
	analysis	simulation	analysis	simulation
6	1.39	1.54	1.02	1.08
12	1.85	2.00	1.42	1.43
18	2.25	2.36	1.81	1.80
24	2.64	2.77	2.20	2.21
30	3.03	3.16	2.59	2.60

TABLE II
TRANSFER TIMES (IN SECONDS) FOR $e = 0.10$

F (KB)	W_{iw} = 1		W_{iw} = 4	
	analysis	simulation	analysis	simulation
6	2.71	2.95	2.92	3.14
12	4.00	4.26	3.98	4.29
18	4.99	5.24	5.28	5.38
24	5.50	5.82	5.61	6.01
30	6.35	6.68	7.18	7.46

relatively large $cwnd$ becomes more beneficial in the random error environment than in the bursty error environment.

B. Effect of the Initial Window

Tables I and II present analytical and simulation transfer times for $W_{\max} = 8$, $l = 125$ ms, $BL = 3$, and various values of F, e, and W_{iw}. From Table I, it can be seen that in a low error rate environment $(e = 0.01)$, WAP 2.0 performs better when $W_{\mathrm{iw}} = 4$ than when $W_{\mathrm{iw}} = 1$. This is because the increase of initial window reduces the number of unnecessary roundtrip times. From Table II, it can be seen that in a high error rate environment $(e = 0.10)$, WAP 2.0 performs better when $W_{\mathrm{iw}} = 1$ than when $W_{\mathrm{iw}} = 4$. To explain this, we observe the number of packet losses and timeouts. It is found that the number of timeouts remains almost the same in both cases ($W_{\mathrm{iw}} = 4$ and $W_{\mathrm{iw}} = 1$), but the number of packet losses is higher when $W_{\mathrm{iw}} = 4$ than when $W_{\mathrm{iw}} = 1$. Since one or few packets can only be recovered per roundtrip time, more time will be needed to transfer a given file when $W_{\mathrm{iw}} = 4$ than when $W_{\mathrm{iw}} = 1$.

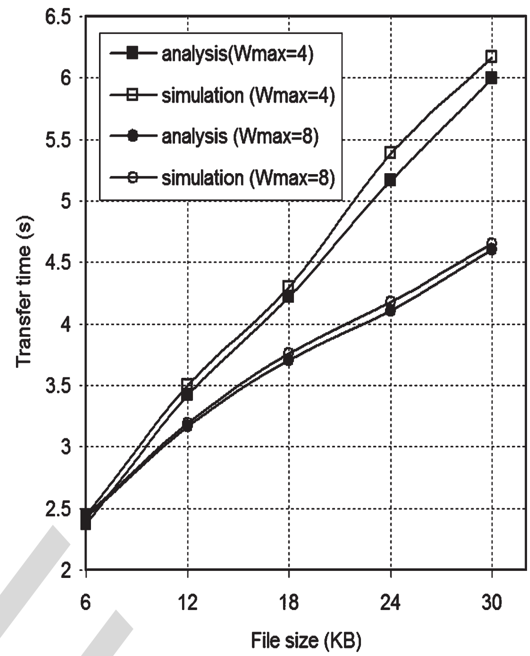


Fig. 6. Transfer time versus F for $e = 0.01$ and $W_{\max} = 4$ and 8.

C. Effect of the Maximum Window Size

The variations of transfer time with file size F are observed at $W_{\max} = 4$ and 8 in a low error rate environment $(e = 0.01)$ and high error rate environment $(e = 0.15)$. In each case, we set $l = 250$ ms, $BL = 1$, and $W_{\mathrm{iw}} = 1$. It is found that as the size of the transferred file increases in a low error rate environment (Fig. 6), the difference between transfer times when $W_{\max} = 4$ and when $W_{\max} = 8$ increases significantly. However, in a high error rate environment (Fig. 7), the increase of the difference between transfer times becomes less significant. This is due to the fact that with a sufficiently large bandwidth delay product in the low error rate environment, $cwnd$ tends to grow to larger values as the size of the transfer file increases. Therefore, $W_{\max}$ becomes a limiting factor for $cwnd$. In the case of high error rate environment, $cwnd$ is mostly limited to low values by the WP-TCP congestion control response. Therefore, $W_{\max}$ becomes less significant in dictating the size of $cwnd$. The transfer time is further studied for different values of l at $W_{\max} = 4$ and 8, in a low error rate environment $(e = 0.01)$, and in a high error rate environment $(e = 0.15)$. In each case, we set $F = 30$ kB, $BL = 1$, and $W_{\mathrm{iw}} = 1$. As expected from previous explanations, the difference in transfer times when $W_{\max} = 4$ and when $W_{\max} = 8$ increases significantly in the low error rate environment (Fig. 8) and slightly in the high error rate environment (Fig. 9) as l increases.

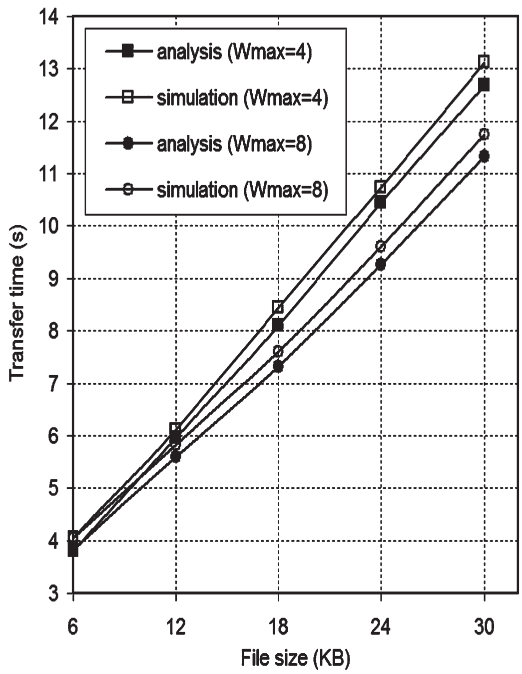


Fig. 7. Transfer time versus F for $e = 0.15$ and $W_{\max} = 4$ and 8.

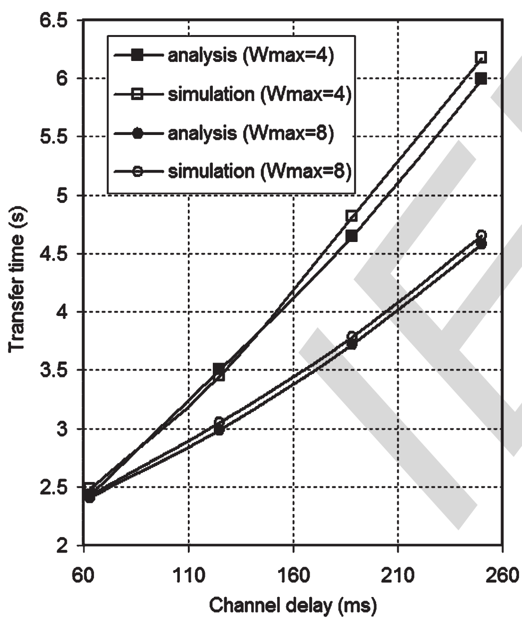


Fig. 8. Transfer time versus l for $e = 0.01$ and $W_{\max} = 4$ and 8.

V. Conclusion

In this paper, an analytical model of the transfer delay of WAP 2.0 for short-lived flows is proposed. Computer simulation results demonstrate that the proposed analytical model produces a good prediction of the WAP 2.0 transfer delay. These results show that for large file sizes ($>$ 30 kB), WAP 2.0 is more sensitive to bursty packet losses than random packet losses. Furthermore, the transfer delay of WAP 2.0 can be improved by increasing the size of the initial window in a low-rate bursty error environment. However, in a high-rate bursty error environment, a larger initial window degrades the transfer delay performance. To improve the transfer delay performance

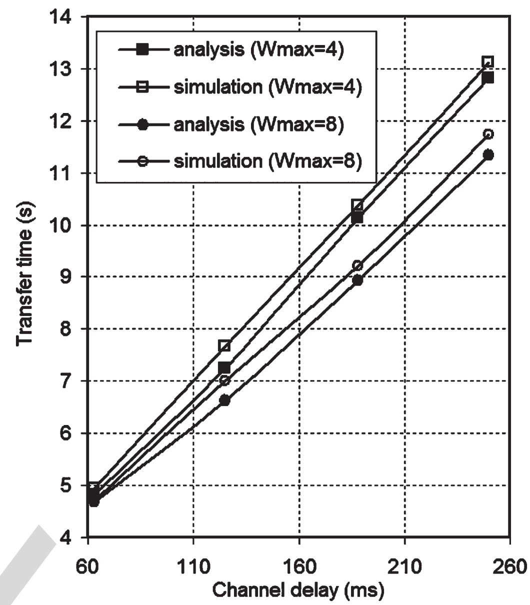


Fig. 9. Transfer time versus l for $e = 0.15$ and $W_{\max} = 4$ and 8.

of WAP 2.0 short-lived flows, a scheme that determines an optimal setting for initial window needs further investigation.

References

[1] WAP Forum. (2001, Jul.). *WAP Architecture Specification*. [Online]. Available: http://www.wapforum.org/what/technical.htm
[2] WAP Forum. (2002, Jan.). *WAP 2.0 Technical White Paper*. [Online]. Available: http://www.wapforum.org
[3] M. Allman, V. Paxson, and W. Stevens, *TCP Congestion Control*, Apr. 1999. RFC 2581.
[4] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, Aug. 1988.
[5] WAP Forum. (2001, Mar.). *Wireless Profiled TCP*. [Online]. Available: http://www.wapforum.org/what/technical.htm
[6] B. A. Mah, "An empirical model of HTTP network traffic," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997, vol. 2, pp. 592–600.
[7] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic pattern and characteristics," *IEEE Netw.*, vol. 6, no. 11, pp. 10–23, Nov. 1997.
[8] H. Rutagemwa and X. Shen, "Modeling and analysis of WAP performance over wireless links," *IEEE Trans. Mobile Comput.*, vol. 2, no. 3, pp. 221–232, Jul.–Sep. 2003.
[9] P. Stuck and C. Hoymann, "Performance evaluation of WAP-based applications over GPRS," in *Proc. IEEE/ICC*, New York, May 2002, pp. 3356–3360.
[10] A. Andreadis, G. Benelli, G. Giambene, and B. Marzucchi, "Performance analysis of the WAP protocol over GSM-SMS," in *Proc. IEEE/ICC*, St. Petersburg, Russia, Jun. 2001, vol. 2, pp. 467–471.
[11] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1742–1751.
[12] M. Mellia, I. Stoica, and H. Zhang, "TCP model for short lived flows," *IEEE Commun. Lett.*, vol. 2, no. 2, pp. 85–87, Feb. 2002.
[13] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 959–971, Dec. 2003.
[14] C. Barakat and E. Altman, "Performance of short TCP transfers," in *Networking*, vol. 1815. New York: Springer-Verlag, 2000, pp. 567–579.
[15] R. Braden, *Requirements for Internet Hosts—Communication Layers*, Oct. 1989. STD 3, RFC 1122.
[16] M. Allman, S. Floyd, and C. Partridge, *Increasing TCP's Initial Window*, Sep. 1998. RFC 2414.
[17] M. Mathis, J. Mahdavi, S. Floyd, and R. Romanow, *TCP Selective Acknowledgment Options*, Oct. 1996. RFC 2018.
[18] M. Zorzi, R. R. Rao, and L. B. Milstein, "On the accuracy of a first-order Markov model for data block transmission on fading channels," in *Proc. IEEE ICUPC*, Tokyo, Japan, Nov. 1995, pp. 211–215.

[19] M. Ross, R. Vicensi, and M. Zorzi, "Accurate analysis of TCP on channels with memory and finite round-trip delay," *IEEE Trans. Wireless Commun.*, vol. 3, no. 2, pp. 627–640, Mar. 2004.

[20] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*. Reading, MA: Addison-Wesley, 2000.

[21] R. A. Howard, *Dynamic Probabilistic Systems*. New York: Wiley, 1971.

[22] S. M. Ross, *Introduction to Probability Models*, 7th ed. Orlando, FL: Harcourt Brace Jovanovich, 2000.

[23] M. Zorzi and R. R. Rao, "Lateness probability of a retransmission scheme for error control on a two-state Markov channel," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1537–1548, Oct. 1999.

[24] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Trans. Netw.*, vol. 8, no. 2, pp. 133–145, Apr. 2000.

AQ2 [25] *The Network Simulator ns-2 Home Page*. [Online]. Available: http://www.isi.edu/nsnam/ns

AQ3 **Humphrey Rutagemwa** (S'XX) received the B.Sc. degree (with first-class honors) in electronics and communications from the University of Dar es Salaam, Dar es Salaam, Tanzania, in 1998 and the M.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002, where he is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering.

From 1999 to 2000, he was a System Engineer with the CRDB Bank Limited. Since 2003, he has been a Research Assistant with the Centre for Wireless Communications, University of Waterloo. His research interests include modeling and performance evaluation of broadband and wireless networks.

Xuemin (Sherman) Shen (M'97–SM'02) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees from Rutgers University, Camden, NJ, in 1987 AQ4 and 1990, respectively, all in electrical engineering.

From September 1990 to September 1993, he was first with the Howard University, Washington, DC, and then the University of Alberta, Edmonton, AB, Canada. Since October 1993, he has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as a Professor and an Associate Chair for Graduate Studies. He is a coauthor of two books and has published more than 200 papers and book chapters on wireless communications and networks and on control and filtering. His research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications systems, wireless security, and *ad hoc* and sensor networks.

Dr. Shen served as the Technical Program Committee Chair for Qshine'05 and as a Co-Chair for the IEEE Broadnet'05, WirelessCom'05, IFIP Networking'05, ISPAN'04, and IEEE Globecom'03 Symposium on Next Generation Networks and Internet. He also serves as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY; *ACM/Wireless Network*; *Computer Networks*; *Dynamics of Continuous, Discrete, and Impulsive—Series B: Applications and Algorithms*; *Wireless Communications and Mobile Computing (Wiley)*; *International Journal Computer and Applications*; and as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and IEEE COMMUNICATIONS MAGAZINE. He received the Premier's Research Excellence Award from the Province of Ontario for demonstrated excellence of scientific and academic contributions in 2003 and the Distinguished Performance Award from the Faculty of Engineering, University of Waterloo, for outstanding contribution in teaching, scholarship, and service in 2002. He is a Registered Professional Engineer in Ontario.

Jon W. Mark (M'62–SM'80–F'88–LF'03) received the B.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1962 and the M.Eng. and Ph.D. degrees from McMaster University, Hamilton, ON, in 1968 and 1970, respectively, all in electrical engineering.

From 1962 to 1970, he was an Engineer and then a Senior Engineer at Canadian Westinghouse Company, Ltd., Hamilton. In September 1970, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, where he is currently a Distinguished Professor Emeritus. He served as the Department Chairman from July 1984 to June 1990. In 1996, he established the Centre for Wireless Communication at the University of Waterloo and is currently serving as its founding Director. He has been on sabbatical leave at the following places: IBM Thomas J. Watson Research Center, Yorktown Heights, NY, as a Visiting Research Scientist (1976–1977); AT&T Bell Laboratories, Murray Hill, NJ, as a Resident Consultant (1982–1983); Laboratoire MASI, Université Pierre et Marie Curie, Paris, France, as an Invite Professor (1990–1991); and the Department of Electrical Engineering, National University of Singapore, Singapore, as a Visiting Professor (1994–1995). He has previously worked in the areas of adaptive equalization, image and video coding, spread spectrum communications, computer communication networks, ATM switches design, and traffic management. He recently coauthored the text entitled *Wireless Communications and Networking* (Prentice-Hall, 2003). His current research interests include broadband and wireless communication, resource and mobility management, and cross-domain interworking.

Dr. Mark received the 2000 Canadian Award for Telecommunications Research and the 2000 Award of Merit of the Education Foundation of the Federation of Chinese Canadian Professionals. He has served as an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS (1983–1990), a member of the Inter-Society Steering Committee of the IEEE/ACM TRANSACTIONS ON NETWORKING (1992–2003), a member of the IEEE Communications Society Awards Committee (1995–1998), an Editor of *Wireless Networks* (1993–2004), and an Associate Editor of *Telecommunication Systems* (1994–2004).

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = Could the term "interworking" be "internetworking" instead, as mentioned in Section 2, paragraph 1?
AQ2 = Please provide additional information in Ref. [25].
AQ3 = Please provide IEEE membership history.
AQ4 = Please check if the city provided is correct. Otherwise, kindly make the necessary modification.

END OF ALL QUERIES