Performance Modeling and Analysis of Window-Controlled Multimedia Flows in Wireless/Wired Networks

Lin Cai, *Member, IEEE,* Xuemin (Sherman) Shen, *Senior Member, IEEE,* Jon W. Mark, *Life Fellow, IEEE,* and Jianping Pan, *Member, IEEE*

***Abstract*— In this paper, we develop a novel analytical framework for modeling and quantifying the performance of window-controlled multimedia flows in a hybrid wireless/wired network. The framework captures the traffic characteristics of window-controlled flows and is applicable to various wireless links and packet transmission schemes. We show analytically the relationship between the sender window size, the wireless link throughput distribution, and the delay distribution. We then substantiate the analysis by demonstrating how to statistically bound the end-to-end delay of flows controlled by a TCP-like Datagram Congestion Control Protocol (DCCP) over an M-state Markovian wireless link. Simulation results validate the analysis and demonstrate the effectiveness and efficiency of the proposed delay control scheme. The scheme can also be applied to other window-based transport layer protocols.**

***Index Terms*— Communication system performance, protocols, modeling, algorithms, resource management, multimedia systems.**

I. INTRODUCTION

WIRELESS cellular networks and the Internet are anticipated to converge into a ubiquitous information transport infrastructure, allowing mobile users to access the IP-based hybrid wireless/wired networks for multimedia services anywhere, anytime. However, there are great challenges to provide *timely* delivery services for delay-sensitive multimedia applications over hybrid networks: 1) wireless channels exhibit inherent severe impairments. For a given wireless channel and Forward Error Correction (FEC) coding scheme, the residual transmission error rate is still visible to the upper layer protocols. To reduce the transmission errors visible to the upper layers, the link layer uses the Automatic Repeat reQuest (ARQ) scheme to detect transmission errors and performs retransmissions locally [1], [2]. For delay-sensitive applications, a low-persistence ARQ scheme is more appropriate: the link layer retransmits corrupt packets a prescribed number of times, and then discards those packets failed to be successfully transmitted thereafter. On the other hand, Channel-State-Dependent (CSD) transmission and scheduling schemes are proposed to achieve multi-user diversity gain and to improve link utilization [3], [4]. With the ARQ and CSD schemes, the wireless link throughput (the number of frames being successfully transmitted over the link per unit time) and delay are random, which bring new challenges to end-to-end delay control [5]; 2) with scalable, error-resilient source coding schemes, emerging Internet-based multimedia applications can adapt to network dynamics, and they require timely delivery and smooth throughput rather than reliability. The currently dominant transport layer protocol, TCP, is not desirable: TCP's congestion control algorithm may halve the sending rate in response to a single packet[1] loss, and its reliable, in-order delivery may introduce excessive delay. Thus, two paradigms of TCP-friendly congestion control algorithms have been proposed [6]–[9] to achieve smoother flow throughput: window-based binomial control and equation-based rate control; based on these control algorithms, a Datagram Congestion Control Protocol (DCCP) has been proposed [10] to provide datagram delivery service for applications which can tolerate a certain degree of packet losses.

To maintain network stability and integrity, delay-sensitive multimedia applications can use the DCCP protocol to regulate their traffic. Specifically, before a packet is sent, the application determines the most appropriate data in that packet; it is up to the transport layer protocol to statistically bound the end-to-end delay of the packet using the services provided by the network, where the end-to-end delay measures the time a packet being sent and received by the transport layer. In this context, a key unsolved issue is whether it is possible for the transport layer protocol to provide timely delivery service over IP-based hybrid networks, and if so, how?

To meet the challenge, the delay performance of cross-domain congestion-controlled flows should be obtained. Our focus is on window-based protocol because window-based congestion control mechanisms are simple to implement, and the *acknowledgment self-clocking* property is desired for delay control over time-varying wireless links. Previous delay analyses of window-controlled TCP flows either assume that the arrival process of TCP packets at the wireless links follows

Manuscript received June 8, 2005; revised January 30, 2006; accepted February 15, 2006. The associate editor coordinating the review of this paper and approving it for publication was Q. Zhang. This work has been supported by a Postgraduate Scholarship and a Strategic Project Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

L. Cai is with the Department of Electrical & Computer Engineering, University of Victoria, Victoria, BC, Canada (email: cai@uvic.ca).

X. Shen (corresponding author) and J. W. Mark are with the Centre for Wireless Communications in the Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (email: {xshen, jwmark}@bbcr.uwaterloo.ca).

J. Pan is with the Department of Computer Science, University of Victoria, Victoria, BC, Canada (email: pan@uvic.ca).

Digital Object Identifier 10.1109/TWC.2007.05449.

[1] Modern transport layer protocols can negotiate maximum segment size on their connection establishment to avoid IP fragmentation. Link layer fragmentation is not considered here, since delay-sensitive multimedia traffic usually has small packet size. In the sequel, the term *packet* is used generically to represent the link *frame*, network *packet*, and transport *segment*.

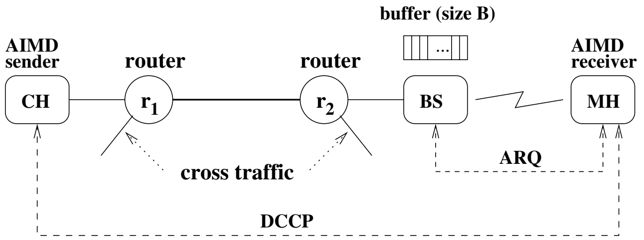


Fig. 1. System model.

a memoryless Markov process [11], [12], or use a fluid model to approximate the packet flow [13], [14]. For a closed-loop window-controlled flow, new packets are injected into the network when *ack*s of previous packets are received by the sender, so the arrival process is highly auto-correlated, and assuming a Markov process for window-controlled flows is impractical in general. A fluid model is only suitable in an environment where the insertion delay is small when compared with the total transfer time and does not vary significantly from packet to packet. Thus, the fluid model approach is not suitable for delay analysis in wireless links with ARQ.

In this paper, a novel analytical framework is developed for modeling and quantifying the performance of window-controlled flows in hybrid wireless/wired networks. We use *token* control to emulate the window-based flow and congestion control. Given the system model and with the token emulation, we obtain the delay distribution and packet loss rate of cross-domain flows as a function of the sender window size and the wireless link throughput distribution. Then, we substantiate the analysis by demonstrating how to bound end-to-end delay over an M-state Markovian wireless link for flows controlled by a TCP-like DCCP protocol [15], [16].

The main contributions of the paper are as follows. An analytical framework is developed to capture the traffic characteristics of closed-loop window-controlled flows, and it is applicable to various wireless links with different channel characteristics and transmission schemes. The analytical results reveal how the window size is related to the delay distribution, which can offer much-needed insights for tuning transport layer protocol parameters to provide timely delivery services in hybrid networks. The analytical results are validated by simulations using the Network Simulator (*ns-2*) [17].

The remainder of the paper is organized as follows. Sec. II presents the system model and the QoS indexes, introduces the TCP-like DCCP protocol, and calculates the wireless link throughput distributions. Based on the system model, in Sec. III, the delay distribution and the packet loss rate of window-controlled flows over hybrid networks are analytically obtained, and a delay control scheme is proposed. Simulation results are given in Sec. IV, followed by further discussion in Sec. V. Concluding remarks and future research issues are given in Sec. VI.

II. System Model

As shown in Fig. 1, we consider a cross-domain multimedia connection between a mobile host (MH) and a correspondent host (CH), through a last-hop wireless link between the MH and a base station (BS). In the wired domain, because of the tremendous advances in optical communications and the fact that a majority of Internet traffic is responsive TCP or TCP-friendly traffic, Internet service guarantee (in terms of delay and loss) has been partly achieved with an engineering solution, bandwidth over-provisioning[2].

However, because the wireless bandwidth is at a premium, bandwidth over-provisioning is not a practical solution in wireless cellular networks. Instead, centralized resource management schemes have been deployed to maintain the QoS of existing and handoff flows. A number of wireless channels with fixed raw capacity are assigned to a number of multimedia flows according to their QoS requirements, since wireless resources are channelized in cellular networks. The allocated resources are infrequently adjusted, when new and handoff flows arrive, or existing flows depart or terminate.

In our system model, the bottleneck of the cross-domain traffic is presumably the wireless link, and end-to-end delay variation mainly happens in the wireless domain. In the following, we will focus on the transmission from the CH to the MH. Our approach is also applicable to the reverse direction transmission.

A. QoS Indexes for Delay-Sensitive Applications

For delay-sensitive multimedia applications over best-effort and highly dynamic networks, scalable source coding schemes with error concealment property have been proposed and are anticipated to be widely deployed, *e.g.*, multiple description (MD) coding schemes [18][3]. With the scalable coding technique, the source encoder determines an optimal bit-rate, according to the current sending rate in the transport layer and the maximum tolerable packet loss rate; thus, the transport layer sender can be assumed saturated, *i.e.,* it always has data to deliver. In addition, for delay-sensitive applications, packets that have suffered excessive delay are useless and will be discarded by the receiver. Therefore, besides flow throughput and packet loss rate, an important QoS index is delay outage rate, which is the ratio of packets with end-to-end delay exceeding a threshold pre-defined by the application. End-to-end delay has a deterministic part and a random part, and the latter is referred to as delay jitter. Delay outage rate equals the ratio of packets with delay jitter exceeding a threshold.

B. TCP-like DCCP Protocol

We focus on the window-based TCP-like DCCP protocol [15], [16], since TCP congestion control is relatively well understood. The flow and congestion control mechanism used by TCP-like DCCP is similar to that in SACK-based TCP [19]: a DCCP sender uses a sender window to limit the maximum number of packets that can be sent out before it gets an acknowledgment back from the receiver, and the

[2]The major ISPs in North America have claimed to guarantee delay jitter in the granularity of milliseconds and packet loss rate $<< 1\%$ in their backbone networks, according to their Service Level Agreements.

[3]An MD coder partitions the source data into several sets and then compresses independently to produce descriptions. The quality of reconstructed multimedia can be improved when more descriptions are received, and the MD decoder can conceal packet losses up to a certain degree. MD coding is very attractive for Internet-based multimedia applications, and it is adopted in our system model.

TABLE I
NOTATIONS

1. γ: received instantaneous signal-to-noise ratio
2. g_i: channel state i
3. π_i: steady-state probability of state i
4. e_i: packet error rate of state i
5. p_e: average packet error rate over the wireless link
6. $P_{i,j}$: state transition probability from state i to state j
7. $N(\Gamma_i)$: level crossing rate at Γ_i
8. λ: the 2nd dominant eigenvalue of channel state transition matrix
9. $T_n(x)$: probability of successfully transmitting x packets in n slots
10. t_s: time to transmit a packet over the wireless link
11. W: sender window size
12. R: round-trip time without queuing delay and retransmission over the wireless link
13. B: buffer size
14. Q: queue length
15. d_q: queuing delay
16. $\boldsymbol{W}$: vector $\{W_1, W_2, \cdots, W_N\}$
17. $T_n(x; P_i)$: probability of successfully transmitting x packets from the i-th flow in n slots, and P_i is the share of bandwidth assigned to the i-th flow
18. S: sum of windows of all flows in the wireless link
19. d_l: local retransmission delay
20. r_t: number of link layer local retransmissions
21. $\tilde{d}_q$: delay in the new arrival queue

sender window size (W) is the minimum of the congestion window (*cwnd*) and the receiver-advertised window (*rwnd*). The sender adjusts the congestion window (*cwnd*) according to the network congestion condition. To avoid a fast sender overrunning a slow receiver, the DCCP receiver advertises the amount of the allocated buffer for the connection, and the sender uses the *rwnd* to bound the amount of in-flight packets. Different from TCP, the DCCP sender is not obligated to retransmit corrupt or lost packets. Due to the space limitation, for other protocol details, the reader is referred to [15], [16].

C. Wireless Link Throughput Distribution

Since the wireless link is presumably the bottleneck, its throughput distribution is closely related to end-to-end delay. Link throughput depends on the channel characteristics, and the transmission and error control schemes used in the physical and link layers.

In a typical wireless channel with multipath propagation and non-line-of-sight (NLOS) frequency-nonselective (flat) fading, the received signal envelope can be modeled as a Rayleigh fading channel. Given the modulation scheme, the channel fading characteristics can be mapped to the packet level. However, the performance analysis of the upper layer protocols becomes quite complex with this approach. Alternatively, a Rayleigh fading channel can be approximated by a finite-state Markov model, which is widely acknowledged as a reasonably accurate and mathematically tractable approach [20], [21]. For easy reference, the notations used throughout the paper are tabulated in Table I.

A finite-state Markov model can be built by discretizing the received instantaneous signal-to-noise ratio (SNR) into several levels. Let Γ_i represent the i-th level. The wireless channel evolves as an M-state ergodic Markov chain $G = \{G_n; n \in \mathbb{Z}_+\}$, where $G_n \in \mathcal{G} = \{g_1, g_2, ..., g_M\}$. Let all packets have the same packet size and assume that the channel remains in one state during a time slot, the transmission time of a packet. The channel is in state g_i if the received SNR is between Γ_i and Γ_{i+1}. For a Rayleigh channel with additive Gaussian noise, the received instantaneous SNR, γ, is exponentially distributed: $\Pr\{\gamma\} = \frac{1}{\gamma_0}\exp(-\gamma/\gamma_0)$, where γ_0 is the mean of the received SNR. The steady state probability of state g_i is $\pi_i = \exp(-\Gamma_i/\gamma_0) - \exp(-\Gamma_{i+1}/\gamma_0)$. Given a modulation scheme and an FEC code, SNR can be mapped to bit error rate and packet error rate. Let e_i be the packet error rate of g_i. The average packet error rate, p_e, equals $\sum_{i=1}^{M} e_i\pi_i$.

Let $P_{i,j}$ be the state transition probability from g_i to g_j. Assume that there is no state transition within a slot, and the state transitions only happen between adjacent states, *i.e.*, $P_{i,j} = 0$ for $|i-j| > 1$. The state transition probabilities can be approximated as [20]

$$P_{i,i+1} \approx N(\Gamma_{i+1})t_s/\pi_i, \quad i = 1, 2, ..., M-1 \quad (1)$$

$$P_{i,i-1} \approx N(\Gamma_i)t_s/\pi_i, \quad i = 2, 3, ..., M, \quad (2)$$

where t_s is the time duration of a slot, and $N(\Gamma_i)$ is the level crossing rate of the Rayleigh fading envelope at Γ_i. $N(\Gamma_i)$ can be calculated according to Γ_i/γ_0 and the Doppler frequency shifts (μ) introduced by mobility: $N(\Gamma_i) = \sqrt{2\pi}\,\mu\,(\Gamma_i/\gamma_0)\,exp\left(-(\Gamma_i/\gamma_0)^2\right)$ [22].

Since the Markov chain is aperiodic and indecomposable, the probability distribution of G_n will converge to the steady-state probability distribution π in a geometrical manner [23]:

$$|\Pr\{G_n = g_i|G_1\} - \pi_i| \leq K \cdot \lambda^n, \quad (3)$$

where K is a constant determined by the initial state G_1, λ is the second dominant eigenvalue of the channel state transition matrix. λ represents the *memory* of a Markov chain. For a typical wireless channel, λ is in the range $[0, 1)$. A channel with positive value of λ means that the probability of remaining in a given state is greater than the stationary probability of the state. A channel with $\lambda = 0$ means that the next state is independent of the current state.

Finding an efficient packet transmission scheme is an active research topic [3], [4]. The optimal transmission scheme depends on λ. In general, if the value of λ is small, a traditional persistent transmission scheme is efficient. If the channel has a large λ, the CSD transmission scheme is more efficient [3]: at the end of the transmission, the transmission may be suspended for some time slots if the channel quality during the past time slot is poor. How long the channel suspends transmission is determined according to λ and the cost of failed transmission [3].

1) Throughput Distribution with Persistent Transmission Scheme: With a persistent transmission scheme, one packet is transmitted per slot no matter what the channel condition is. The persistent scheme is used when λ approaches zero, and the probability of successfully transmitting x packets in n slots is

$$T_n(x) = \binom{n}{x}\left(\sum_{i=1}^{M}\pi_i(1-e_i)\right)^x \left(\sum_{i=1}^{M}(\pi_i e_i)\right)^{n-x}. \quad (4)$$

When n is large, T_n can be approximated by a Gaussian random variable with mean $E[T_n] = (1-p_e)n$ and variance $Var(T_n) = (1-p_e)p_e n$.

2) Throughput Distribution with CSD Transmission Scheme: With a CSD transmission scheme, to calculate the throughput distribution in n slots, the probability and throughput of each possible state trajectory, $\{G_1, G_2, \cdots, G_n\}$, should be calculated. $T_n(x)$ is the sum of the probabilities of those trajectories with throughput equal to x packets. The number of trajectories increases exponentially with n, and so does the computational complexity. To simplify the computation, we further explore the convergence speed of the Markovian channel to obtain the approximation of the throughput distribution in polynomial time.

If $|\Pr\{G_k|G_1\} - \pi|$ is smaller than a small positive value ϵ, the channel states after the k-th slot are assumed independent of G_1. Thus, n slots can be divided into n/k blocks, and each block has k consecutive slots. The number of packets being successfully transmitted in n slots, T_n, is approximately the sum of n/k i.i.d. random variables T_k (number of successful transmissions in k slots). Thus, the probability distribution of T_n can be approximated as the convolution of the distribution of T_k for $n/k - 1$ times, where k is the minimal integer satisfying $|\Pr\{G_k = i|G_1\} - \pi_i| \leq \epsilon$. According to (3), k is proportional to $\log_\lambda \epsilon$. When $n >> k$, T_n can be further simplified as a Gaussian random variable with mean $(n/k)\mathrm{E}[T_k]$ and variance $(n/k)\mathrm{Var}(T_k)$, and the computational complexity for throughput distribution is only $O(M^k)$.

III. Performance Analysis

To maintain fairness and network stability, window-based transport layer protocols adjust *cwnd* according to network conditions. Window dynamics for congestion control have been extensively studied. On the other hand, end-to-end delay is mainly related to the window size during the past round-trip time (*rtt*). Since congestion control and delay control are of different time-scale, they can be studied separately. We first focus on the delay performance with a fixed window size. Then, we substantiate the analysis by developing a delay control scheme for DCCP-controlled flows.

We make the following *assumptions*, and the impacts of the assumptions will be discussed in Sec. V: a1) let R slots denote the *rtt without* queuing delay and retransmission over the wireless link. R can be approximately constant, since wired links are less dynamic than wireless ones, and delay jitter in the wireless domain is dominant and can absorb delay jitter in the wired domain; a2) the receiver acknowledges each received packet without delay, and *ack*s can be successfully transmitted without loss; a3) the sender is saturated; a4) the sender sets round-trip timers conservatively enough to allow sufficient time for the low-persistence ARQ scheme to recover from transmission errors. This assumption tends to hold, since the timeout value is estimated as the average of measured *rtt*s plus a conservative factor proportional to the measured standard deviation. Furthermore, coarse-grained timers with a granularity of 500 ms are implemented in practice; and a5) the probability of packet losses due to failed retransmissions by the link-level ARQ is negligible.

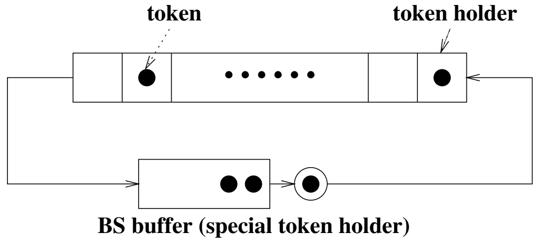


Fig. 2. Token emulation, single flow.

A. Single Flow, Sufficient Buffer

We first consider one window-controlled flow with a sender window size W occupying a wireless link with dedicated buffer. Let the interface buffer size B be no less than W, so that the BS buffer will not overflow. After sending out a window of packets, the sender can only resume sending when it receives a new *ack*. To capture the relationship between *ack*s and data packets, we use *token* control to emulate the window control. A packet, which can be data or *ack*, needs a token to traverse the network. When the receiver receives a data packet with an accompanying token, an *ack* is returned with the token. The sender generates and adjusts the total number of tokens according to the flow and congestion control mechanisms. When the sender window size is W, it is equivalent to having W tokens circulating along the path.

Without queuing delay and retransmission at the BS, a token takes R slots to finish a round-trip. The round-trip in the wired domain can be represented by $R - 1$ token holders. A token shifts to the next holder along the path per time slot. The BS buffer is represented by a special token holder. If a packet is transmitted successfully over the wireless link, the token with that packet can shift from the BS buffer to the next token holder, as shown in Fig. 2. The number of tokens in the $R - 1$ non-special token holders equals the number of successful transmissions over the wireless link during the past $R - 1$ slots; the remaining tokens are queued in the BS buffer, the special token holder.

Proposition 1: For a flow with window size W occupying a wireless link with sufficient buffer ($B \geq W$), if during the past $R - 1$ slots, n packets were transmitted successfully over the wireless link, the BS queue length, Q, would equal $(W - n)^+$, where $0 \leq n \leq R - 1$ and $(x)^+ = \max(x, 0)$.

To fully utilize the wireless resource, (*i.e.*, the wireless link should not be idle whenever the link layer determines to transmit), Q should be greater than zero all the time. In other words, the sufficient condition to fully utilize the wireless resource is $W \geq R$. From Proposition 1, when a packet arrives at the BS buffer, the probability of $Q = x$ is the probability that $W - x$ packets were successfully transmitted over the wireless link during the past $R - 1$ slots, denoted as $T_{R-1}(W - x)$, which has been derived in Sec. II-C.

The time duration from the instant a token leaves the BS buffer to the instant it leaves the BS buffer again, $R + d_q$, equals the time that W packets are successfully transmitted over the wireless link. Therefore, the probability of $R + d_q > R + D$ equals the probability of less than W packets being successfully transmitted over the wireless link in $R + D$ slots.

Proposition 2: For a flow with window size W ($W \geq R$)

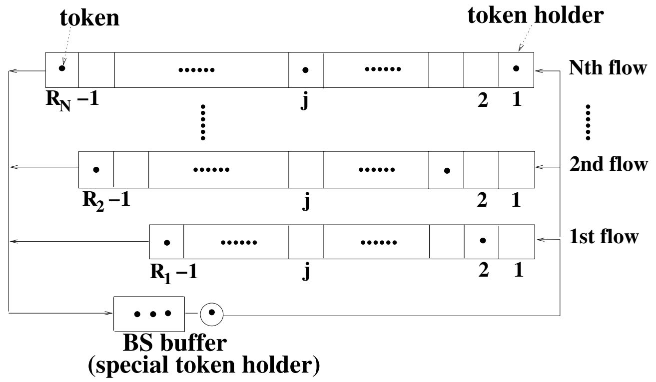


Fig. 3. Token emulation, multiple flows.

occupying a wireless link with buffer size B ($B \geq W$), the probability of queuing delay, d_q, exceeding a threshold D equals the probability of less than W packets being successfully transmitted over the wireless link in $R+D$ slots: $\Pr\{d_q > D|W\} = \sum_{x=0}^{W-1} T_{R+D}(x)$.

When $B \geq W \geq R$, there is no packet loss due to BS buffer overflow, and the flow throughput equals the average link throughput, (*e.g.*, it equals $1-p_e$ packet per slot with the persistent transmission scheme).

B. Single Flow, Limited Buffer

When $B < W$, buffer overflow may occur. With a limited buffer B ($B < W$) and a window size W ($W \geq R$), the probability of buffer overflow equals the probability that less than $W - B$ successful transmissions over the wireless link in the past $R-1$ slots:

$$\Pr\{Q = B|W; B\} = \sum_{x=0}^{W-B-1} T_{R-1}(x). \tag{5}$$

When $W \geq R$, the minimum queue length is $W - R + 1$, and the queue length distribution is $\Pr\{Q = x|W; B\} = T_{R-1}(W-x)$, where $W - R + 1 \leq x < B$.

When the queue length equals x, the probability of queuing delay exceeding D slots is $\sum_{i=0}^{x-1} T_{D+1}(i)$. The queuing delay distribution is given by:

$$\Pr\{d_q > D|W; B\} = \sum_{x=W-R+1}^{B} \sum_{i=0}^{x-1} T_{D+1}(i) \cdot T_{R-1}(W-x). \tag{6}$$

Equations (5) and (6) reveal the conflicting requirements in choosing the value of buffer size: a larger buffer may reduce the probability of buffer overflow at the cost of a larger delay outage rate. To choose a proper buffer size, the cost of transmitting a packet suffering excessive delay and the cost of packet loss due to buffer overflow should be considered.

C. Multiple Flows

When several window-controlled flows share a wireless link, the token emulation approach is still applicable. Fig. 3 shows N flows sharing a wireless link. For the i-th flow, the minimal *rtt* and sender window size are denoted as R_i and W_i, respectively. Without loss of generality, let $R_1 \leq R_2 \leq \cdots \leq R_N$. Denote the probability of BS transmitting packets from the i-th flow as P_i. The mean number of the i-th flow's tokens in the BS buffer equals $W_i - \mathrm{E}[T_{R_i-1}]P_i$, where $\mathrm{E}[T_{R_i-1}]$ is the mean number of successful transmissions over the wireless link in R_i-1 slots. On the other hand, P_i equals the ratio of the i-th flow's tokens in the buffer, *i.e.*, $P_i = \frac{W_i - \mathrm{E}[T_{R_i-1}]P_i}{S - \sum_{j=1}^{N} \mathrm{E}[T_{R_j-1}]P_j}$, where $S = \sum_{i=1}^{N} W_i$. Therefore, to assign the i-th flow a P_i portion of the wireless bandwidth, W_i should satisfy the following condition:

$$W_i = \frac{P_i W_j}{P_j} + (\mathrm{E}[T_{R_i-1}] - \mathrm{E}[T_{R_j-1}])P_i, \quad \text{for} \quad 1 \leq i, j \leq N. \tag{7}$$

Equation (7) gives the *necessary* and *sufficient* condition of proportional fairness for window-controlled flows. According to (7), for $P_i = 1/N$, flows' window sizes are not proportional to their *rtt*s. To achieve fairness for flows with different *rtt*s, the number of coexisting flows and their *rtt*s should be considered. The simulation results in Sec. IV-B validate (7).

Denote $T_n(x; P_i)$ as the probability of successfully transmitting x packets of the i-th flow in n slots. $T_n(x; P_i) = \sum_{k=0}^{n-x} \binom{x+k}{x} P_i^x (1-P_i)^k T_n(x+k)$, where $0 < P_i < 1$. Obviously, $T_n(x) = T_n(x; 1)$, and $T_n(x; P_i) + T_n(x; P_j) = T_n(x; P_i + P_j)$. To fully utilize the wireless link, the sufficient condition is $S \geq R_N$.

Let $B \geq S \geq R_N$ (sufficient buffer case). The number of i-th flow's tokens outside the BS buffer equals the number of i-th flow's packets being successfully transmitted during the past $R_i - 1$ slots. Thus, the queue length distribution is

$$\Pr\{Q = x|\boldsymbol{W}\} = \sum \cdots \sum_{y_1+\cdots+y_N=S-x} T_{R_1-1}(y_1; 1) \cdots T_{R_N-R_{N-1}}(y_N; P_N), \tag{8}$$

where vector $\boldsymbol{W} = \{W_1, \cdots, W_N\}$, and $S - R_N + 1 \leq x \leq S$.

The probability of queuing delay exceeding D slots is

$$\Pr\{d_q > D|\boldsymbol{W}\} = \sum_{x=S-R_N+1}^{S} \Pr\{Q = x|\boldsymbol{W}\} \sum_{j=0}^{x-1} T_{D+1}(j), \tag{9}$$

where $S \geq R_N$.

When $B < S$, the buffer overflow probability is $\Pr\{Q = B|\boldsymbol{W}; B\} = \sum_{x=B+1}^{S} \Pr\{Q = x|\boldsymbol{W}\}$. The probability of queuing delay exceeding D slots is given by

$$\Pr\{d_q > D|\boldsymbol{W}; B\} = \sum_{x=S-R_N+1}^{B} \Pr\{Q = x|\boldsymbol{W}\} \sum_{j=0}^{x-1} T_{D+1}(j), \tag{10}$$

where $S \geq R_N$.

D. Local Retransmission Delay

In Secs. III-A - III-C, we assumed that local acknowledgment is received instantaneously after transmission, and local retransmission is triggered immediately afterward. In some situations, there may be a non-negligible delay between a packet being transmitted and its retransmission being triggered. To reduce delay jitter, a non-preemptive priority queuing discipline is used at the BS, such that retransmitted packets have a higher priority. The BS delay jitter contains two parts: queuing delay (in both the new arrival queue and the high-priority retransmission queue) and local retransmission

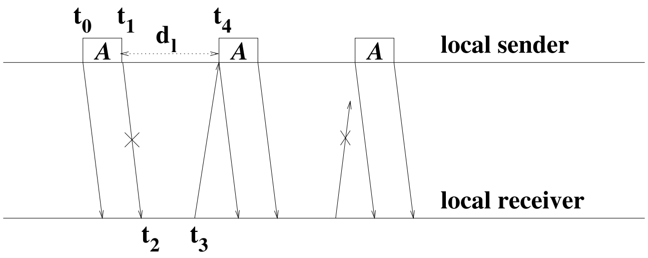


Fig. 4. Local retransmission delay.

delay. This section analyzes the effect of local retransmission delay on end-to-end delay.

As shown in Fig. 4, packet $\mathcal{A}$ is transmitted over the wireless link from t_0 to t_1. At t_3, the local receiver sends a local negative acknowledgment back to the local sender. At t_4, since the local sender receives a local negative acknowledgment or it fails to receive a local acknowledgment, packet $\mathcal{A}$ is retransmitted. We define local retransmission delay d_l as the time duration between two consecutive transmissions of the same packet. Let $\tilde{d_q}$ be the waiting time in the new arrival queue; and r_t be the number of retransmissions. The total delay jitter d_q equals $\tilde{d_q} + r_t(d_l + 1)$.

For a single flow with $B \geq W \geq R$, the probability of delay jitter exceeding threshold D can be calculated as follows:

$$\Pr\{d_q > D|W\} = \sum_{i=W-R+1}^{D-1} \Pr\{r_t > \tfrac{D-i-1}{d_l+1}|\tilde{d_q} = i; W\} \times \Pr\{\tilde{d_q} = i|W\} + \Pr\{\tilde{d_q} > D|W\}. \quad (11)$$

Generally, r_t and $\tilde{d_q}$ are independent random variables. r_t can be simplified as a geometrically distributed random variable. The packet at the front of the new arrival queue will be transmitted only if the retransmission queue length is less than d_l; therefore, $\Pr\{\tilde{d_q} > D|W\}$ is equivalent to the probability of fewer than $W - d_l - 1$ packets are successfully transmitted in $R + D$ slots. Equation (11) can be re-written as

$$\begin{aligned}\Pr\{d_q > D|W\} \;&=\; \sum_{x=0}^{W-d_l-2} T_{R+D}(x) \\ &+ \sum_{i=W-R+1}^{D-1} \left(1 - \sum_{k=0}^{\lfloor \frac{D-i-1}{d_l+1} \rfloor} p_e^k (1-p_e)\right) \\ &\times \sum_{j=0}^{W-d_l-2} \left(T_{R+i-1}(j) - T_{R+i}(j)\right) \quad (12)\end{aligned}$$

Similarly, when $B < W$

$$\Pr\{d_q > D|W; B\} = \Pr\{\tilde{d_q} > D|W; B\} + \sum_{i=W-R+1}^{D-1} \Pr\{r_t > \tfrac{D-i-1}{d_l+1}|W; B\} \Pr\{\tilde{d_q} = i|W; B\}, \quad (13)$$

where

$$\Pr\{\tilde{d_q} > D|W; B\} = \sum_{x=W-R+1}^{B} \sum_{i=0}^{x-d_l-2} T_{D+1}(i) T_{R-1}(W - x).$$

TABLE II

ALGORITHM TO DETERMINE $MaxRwnd$, SINGLE FLOW CASE

```
1  for (w = 0; w ≤ R + D; w + +) {
2      Delay_Outage_Probability+ = T_{R+D}(w)
3      if (Delay_Outage_Probability > Threshold) {
4          MaxRwnd = w − 1
5          break }
6  }
```

For multiple flows case,

$$\Pr\{d_q > D|\boldsymbol{W}; B\} = \Pr\{\tilde{d_q} > D|\boldsymbol{W}; B\} + \sum_{i=S-R_N+1}^{D-1} \Pr\{r_t > \tfrac{D-i-1}{d_l+1}|\boldsymbol{W}; B\} \Pr\{\tilde{d_q} = i|\boldsymbol{W}; B\}, \quad (14)$$

where $S \geq R_N$ and

$$\Pr\{\tilde{d_q} > D|\boldsymbol{W}; B\} = \sum_{x=S-R_N+1}^{S} \Pr\{Q = x|\boldsymbol{W}\} \sum_{j=0}^{x-d_l-2} T_{D+1}(j).$$

E. Rwnd-Based Delay Control for DCCP Protocol

Based on the above analysis, the delay outage probability is a non-decreasing function of the window size. On the other hand, the wireless link is under-utilized when the arrival rate to the BS is less than the average link throughput, so the window size should be large enough for efficient link utilization. Thus, suitable window size is a tradeoff between delay performance and link utilization, and we need to determine the maximum *rwnd* ($MaxRwnd$) which can efficiently utilize wireless link and guarantee delay outage probability.

The BS allocates the wireless channel and calculates the link throughput distribution, $T_{R+D}(x)$, as illustrated in Sec. II-C. For the single flow case, the BS determines $MaxRwnd$ which bounds the delay outage probability. The pseudo code is shown in Table II. When local retransmission delay is considered, the delay outage probability calculated in `Line 2` should be replaced according to (12). Then, the BS sets the buffer size $B = MaxRwnd$, and informs the DCCP sender and receiver to set *rwnd* and *ssthresh* equal to $MaxRwnd$.

For the multiple flows case, the algorithm to determine $MaxRwnd$s is shown in Table III. Since window size is always an integer, W_i should be rounded to the nearest integer, as shown in `Line 3`. The BS then informs DCCP senders and receivers to set the *rwnd*s and *ssthresh*s according to $MaxRwnd$s, and it sets the BS buffer size $B = \sum_{i=1}^{N} MaxRwnd_i$.

Using *rwnd* to enhance TCP performance had been proposed in the literature [24]–[26]. In [24], *rwnd* is used to enhance fairness and reduce packet losses in the wired link. Here, we set the *rwnd* not only to efficiently utilize the time-varying wireless link, but also to statistically bound the delay. In [25] and [26], *rwnd* is used to enhance TCP performance for hybrid ATM/IP and 3G wireless/IP networks, by frequently adjust *rwnd* according to the queue length at the interface node. Since the *rwnd* is determined with the assumption that there is no delay and loss in the wired domain, the performance of the schemes in [26] degrades when the *rtt* is above 100 ms or the packet loss rate in the wired domain

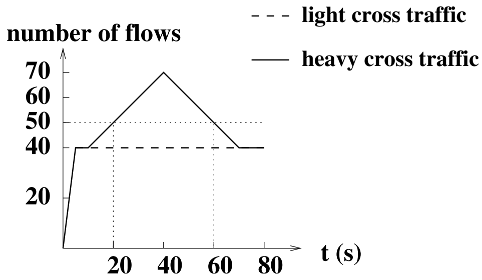


Fig. 5. Number of cross TCP connections in r_1r_2.

exceeds 0.1%. In this paper, instead of frequently changing *rwnd* according to the current queue length, we derive the *rwnd* considering the link profile, flow *rtt*, and application QoS requirements, and the simulation results in Sec. IV show the robustness of our approach.

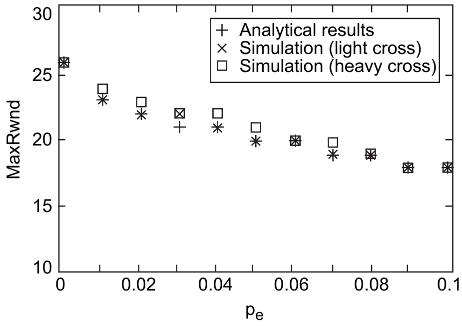
IV. Simulation Results

To verify the analysis, we have performed extensive simulations using ns-2. The simulation topology is the same as that in Fig. 1. For the target DCCP-controlled multimedia flow, the sender is at the CH, and the receiver is at the MH. Cross traffic sharing r_1r_2 are TCP-SACK flows with packet size 1000 bytes. The following parameters are used in the simulation unless otherwise explicitly stated. Wired links between the CH, r_1, r_2, and the BS are duplex links with 100 Mbps and propagation delay 10 ms, 12.5 ms, 10ms, respectively. Both r_1 and r_2 are Random Early Detection (RED [27]) capable. The downlink and uplink between the BS and the MH have capacity 200 Kbps and 100 Kbps, respectively, and propagation delay 5 ms. The downlink channel condition is dynamically changed according to the M-state Markov model. The BS buffer size is set to *rwnd* to avoid buffer overflow, and the BS uses Droptail queue management. The target multimedia flow has packet size 125 bytes. Thus, the duration of a time slot is 5 ms. The *rtt* of the target flow without BS queuing delay is 85 ms. To eliminate the phase effects [28], the *rtt*s of cross TCP traffic are slightly different, between 80 ms and 90 ms. Each simulation lasts 80 seconds, and different initial randomization seeds are used to reduce simulation dynamics. To eliminate system warming-up effects, simulation results for the first 5 seconds are not counted.

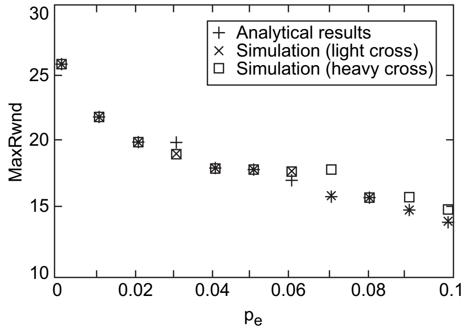
A. Single Flow

To examine the effectiveness of the analysis in various scenarios, simulations with light cross traffic and heavy cross traffic have been performed separately. The number of cross TCP-SACK connections changes according to Fig. 5. In the light cross traffic case, the bottleneck for the target DCCP flow is always the wireless link; in the heavy cross traffic case, the bottleneck is the backbone link between 20 s and 60 s when there are more than 50 TCP connections in r_1r_2.

With the persistent transmission scheme, Fig. 6(a) compare the analytical and simulation results of the maximum *rwnd* ($MaxRwnd$) with which the delay outage rate (ratio of received packets with end-to-end delay jitter exceeding 50 ms) is below 1%. Since the persistent transmission scheme is suitable for channel with small λ, the wireless channel condition evolves following a two-state Markov model with λ equal to zero, and the average packet error rate (p_e) varies from 0% to 10%. The local retransmission delay is 5 ms.



(a) persistent transmission



(b) CSD transmission

Fig. 6. $MaxRwnd$.

TABLE III

Algorithm to Determine $MaxRwnd$, N-Flow Case

```
for (w_1 = 0; w_1 < R_1 + D; w_1++) {
    for (i = 2; i ≤ N; i++)
        w_i = round{P_i(w_1/P_1 + E[T_{R_i-1}] − E[T_{R_j-1}])}
    obtain Delay_Outage_Probability w.r.t. (9)
    if (Delay_Outage_Probability ≤ Threshold)
        for (i = 1; i ≤ N; i++) MaxRwnd_i = w_i
    else break
}
```

The analytical and simulation results with the CSD transmission scheme are shown in Fig. 6(b). The wireless channel evolves following a three-state Markov model with $\lambda = 0.35$. According to the CSD transmission scheme in [3], the BS suspends transmission for one slot if the previous transmission fails, and there is no local retransmission delay.

The analytical $MaxRwnd$ is calculated according to the algorithm in Table II. As shown in Fig. 6, the analytical results approximate the simulation ones well when the cross traffic is light (when the bottleneck is always the wireless link). The analytical ones are slightly more conservative when the cross traffic is heavy. With heavy cross traffic, firstly, the bottleneck link is the backbone link between 20 s and 60 s, so that *cwnd* and W are smaller than *rwnd* during this period, and the delay outage rate will be smaller. Secondly, when r_1r_2 becomes more congested, queuing delay in the wired domain becomes more significant. These two effects offset each other, so that the heavy and light cross traffic cases have the similar delay outage rates.

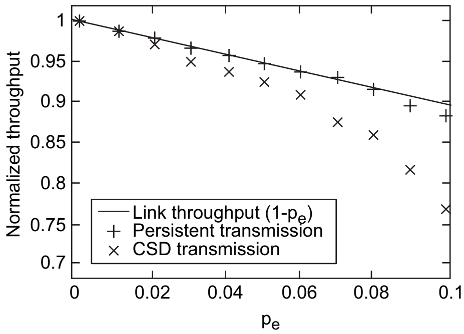


Fig. 7. One DCCP flow, light cross traffic.

Another observation is that $MaxRwnd$ changes slowly w.r.t. p_e. For instance, when p_e varies from 5% to 8%, $MaxRwnd$ only changes from 20 to 19 for the persistent transmission cases, and from 18 to 16 for the CSD transmission cases. Therefore, the *rwnd*-based control scheme can tolerate certain degree of errors in estimating the wireless channel profile.

Comparing Figs. 6 (a) and (b), the $MaxRwnd$ with the CSD transmission scheme is smaller than that with the persistent transmission scheme. This is because the CSD transmission scheme makes the following tradeoff — achieving multi-user diversity gain and higher power efficiency at the cost of possible lower throughput of a particular flow. In our simulations, only a single wireless link is simulated, and the interferences between wireless links are not addressed. Therefore, the multi-user diversity gain is not reflected in the simulation results, and the throughput with the CSD transmission scheme is less than that with the persistent transmission scheme, as shown in Fig. 7. Since our focus is on the performance of the transport layer protocol, we have not explored the multi-user diversity gain here.

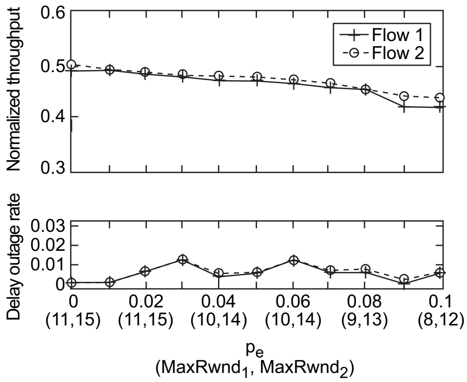
For the link utilization, $MaxRwnd$ should be larger than $R(1 - p_e)$ in order to efficiently utilize the wireless link. Our analytical and simulation results show that $MaxRwnd$ is larger than $R(1 - p_e)$ in most cases except for the CSD transmission scheme with $p_e = 0.1$. Therefore, $MaxRwnd$ does not limit the flow throughput to under-utilize the wireless link in most cases. Fig. 7 also demonstrates the efficiency of the *rwnd*-controlled scheme: with the persistent transmission scheme, by setting *rwnd* to the calculated $MaxRwnd$, the normalized flow throughput (packet per slot) is close to the average link throughput.

In summary, no matter what the link layer transmission scheme is used, it is feasible to calculate $MaxRwnd$ beforehand to bound the delay outage rate. In the following, we only present simulation results with persistent transmission scheme due to space limitations.

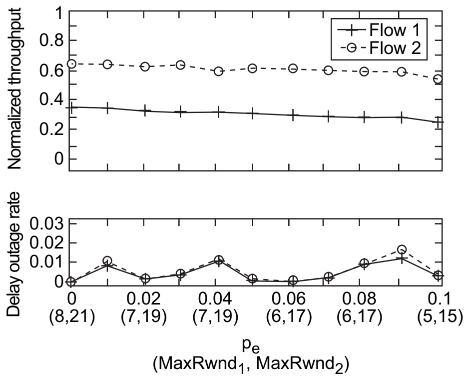
B. Multiple Flows

For N DCCP-flows sharing the wireless link case, the senders are at N CHs which are connected to r_1, and the receivers are at the MH.

Firstly, let two DCCP flows have the same share of the wireless link. Their $MaxRwnd$s are calculated to guarantee that their delay outage rates are below 1%, using the algorithm in Table III. Fig. 8(a) shows the normalized throughputs and



(a) $R_1 = 17$, $R_2 = 25$, $P_1 = P_2 = 1/2$



(b) $R_1 = 17$, $R_2 = 25$, $P_1 = 1/3$, $P_2 = 2/3$

Fig. 8. Two DCCP flows.

delay outage rates of two DCCP flows which have minimal *rtt*s of 17 slots and 25 slots, respectively. The simulation results show that the co-existing DCCP flows can fairly share the wireless link and satisfy the delay outage bounds, no matter whether they have the same *rtt*s or not.

Secondly, let two DCCP flows have different shares of the wireless link. In Fig. 8(b), flow 1 is assigned $1/3$ share of the wireless link, and flow 2 is assigned $2/3$ share. The simulation results show that the co-existing DCCP flows can achieve their fair shares and satisfy the delay outage bounds, even when they have been assigned different shares of the wireless link. The same conclusion can be drawn when more flows sharing the wireless link.

Since $MaxRwnd$s are rounded to integers, the results deviate slightly from the designed target due to quantization errors, *e.g.*, in some cases the flow throughputs have small difference from their shares of the bandwidth, or the delay outage rates slightly exceed the desired threshold. Nevertheless, such deviations can be anticipated and controlled.

V. Further Discussion

In this section, we discuss the assumptions used in Sec. III.

Delay Variation in Wired Network — Delay over the wired domain may also be time-varying. For single flow, let R and d_q be the minimal *rtt* and the BS queuing delay used in the calculation; let R' and d_q' be the actual values experienced by a token. When $R' > R$, there are two cases. Case (a), when the token returns to the BS, the BS queue is non-empty. In this case, the actual round trip delay $(R' + d_q')$ of the token is equal to the time to transmit W packets successfully over the wireless link, *i.e.*, $\Pr\{d'_q + R' > R + D|W\} = \sum_{x=0}^{W-1} T_{R+D}(x) = \Pr\{d_q > D|W\} = \Pr\{d_q + R > R+D|W\}$. Thus, the actual end-to-end delay jitter $R'+d'_q-R$ has the same distribution as that of d_q, so our analytical results are still valid. It is worth to point out that d'_q and R' are *dependent* random variables, so the probability distribution of end-to-end delay is *not* a convolution of the distributions of d'_q and R'. Case (b), when the token returns to the BS, the BS queue is empty. In this case, the delay variation in the wired domain cannot be absorbed by the BS queue. Since the delay in highly-multiplexed wired networks is not under the control of the end-systems, and the Internet measurement results show that the delay jitter over the backbone networks is not significant [29], case (b) is not considered in this paper.

Delayed and Lost Acknowledgments — DCCP-2 protocol defines an Ack Ratio ($AckRatio$), such that the receiver sends one *ack* packet for $AckRatio$ data packets received [16]. The default $AckRatio$ is two, which is similar to TCP's delayed *ack*s. The disturbance of delayed *ack*s to delay distribution is equivalent to the case that R' experienced by some packets is slightly larger than R, which has been discussed in the previous paragraph.

For the lost *ack* case, since *ack*s of the DCCP protocol have an Ack Vector, and a single Ack Vector can acknowledge a number (up to 16192) of data packets. With the redundancy in *ack*s, the sender can recover the information of an occasional lost *ack* from the following *ack*s. Thus, the disturbance of lost *ack*s is equivalent to that of delayed *ack*.

Unsaturated Sender — With unsaturated sender, the number of in-flight packets may be less than W, so that the actual end-to-end delay outage rate may be less than the analytical results, and the *rwnd*-based delay control scheme can still bound the delay outage rate.

Channel Model and Physical/Link Layer Protocols — We use the M-state Markov model and the persistent and CSD transmission schemes to calculate the wireless link throughput distribution. Since the end-to-end performance is given as a function of the link throughput distribution, no matter what the lower layer protocols are used, and no matter what fading characteristics of the wireless channel are, for the upper layer protocol, only the link throughput distribution needs to be re-calculated or measured, and the proposed analytical framework and delay control scheme is still applicable.

In this paper, we focus on the fast-fading characteristics of the wireless channel. End-to-end flow/congestion control for wireless networks with slow-fading or shadowing channels has been extensively investigated in the literature. Since the channel condition is relatively stable for several *rtt*s with slow-fading or shadowing, the BS can notify the CH about the channel condition and the sender can adjust its window size accordingly [30], [31]. However, for fast-fading channels, the channel condition may change drastically within one *rtt*, so the notification schemes cannot be applied directly, and our approach is more effective.

VI. Conclusions and Future Work

We have developed a novel analytical framework for modeling and quantifying the QoS performance of window-controlled flows in hybrid wireless/wired networks. The obtained analytical results should provide meaningful guidelines for system parameter selection to support delay-sensitive multimedia applications over the hybrid networks. Extensive simulations with NS-2 have been performed to validate the analysis and demonstrate the effectiveness and efficiency of the proposed delay control scheme. Our future work will use the traces of real wireless links to verify the analysis. In addition, how to extend the analysis to multi-hop and random accessed wireless links are under investigation.

References

[1] G. Fairhurst and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)," IETF RFC 3366, 2002.
[2] E. Esteves, "The high data rate evolution of the CDMA2000 cellular system," *Multiaccess, Mobility and Teletraffic for Wireless Communications*, vol. 5, pp. 61–72, 2000.
[3] D. Zhang and K. M. Wasserman, "Transmission schemes for time-varying wireless channels with partial state observations," in *Proc. IEEE Infocom'02*, vol. 2, pp. 467–476.
[4] P. Bhagwat, P. P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," in *Proc. IEEE Infocom'96*, pp. 1133–1140.
[5] M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in *Proc. ACM Mobicom'02*, pp. 71–82.
[6] S. Floyd, M. Handle, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM'2000*, pp. 43–56.
[7] Y. R. Yang and S. S. Lam, "General AIMD congestion control," Technical Report TR-2000-09, University of Texas, May 2000. (A shorter version appeared in the Proceedings of ICNP'00, Osaka, Japan, November 2000.
[8] L. Cai, X. Shen, J. Pan, and J. W. Mark, "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 339–355, April 2005.
[9] D. Bansal and H. Balakrishnan, "TCP-friendly congestion control for real-time streaming applications," Technical Report MIT-LCS-TR-806, MIT, May 2000.
[10] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," November 2004 (work in progress). Available at `http://www.icir.org/kohler/dcp/draft-ietf-dccp-spec-09.txt`.
[11] H. M. Chaskar, T. V. Lakshman, and U. Madhow, "TCP over wireless with link level error control: analysis and design methodology," *IEEE/ACM Trans. Networking*, vol. 7, no. 5, pp. 605–615, 1999.
[12] A. Chockalingam, M. Zorzi, and R. R. Rao, "Performance of TCP on wireless fading links with memory," in *Proc. IEEE ICC'98*.
[13] K. W. Fendick, M. A. Rodrigues, and A. Weiss, "Analysis of a rate-based feedback control strategy for long haul data transport," *Performance Evaluation*, vol. 16, pp. 67–84, 1992.
[14] R. Roy, M. Raghuraman, and S. S. Panwar, "Analysis of TCP congestion control using a fluid model," in *Proc. IEEE ICC'01*, pp. 2396–2403.
[15] L. Cai, X. Shen, J. W. Mark, and J. Pan, "QoS support for multimedia traffic in wireless/wired networks using TCP-friendly AIMD protocol," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 469–480, Feb. 2006.
[16] S. Floyd and E. Kohler, "Profile for DCCP congestion control ID 2: TCP-like congestion control, November 2004 (work in progress). Available at `http://www.icir.org/kohler/dcp/draft-ietf-dccp-ccid2-08.txt`.
[17] S. Floyd and S. McCanne, Network Simulator, LBNL public domain software. Available via ftp from `ftp.ee.lbl.gov`. NS-2 is available in `http://www.isi.edu/nsnam/ns/`.
[18] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, pp. 74–93, 2001.
[19] E. Blanton, M. Allman, K. Fall, and L. Wang, "A conservative selective acknowledgment (SACK)-based loss recovery algorithm for TCP," IETF RFC 3517, 2003.
[20] H. S. Wang and N. Moayeri, "Finite-state Markov channel–a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, 1995.
[21] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, 1999.

[22] J. W. Mark and W. Zhuang, *Wireless Communications and Networking*. Prentice Hall, 2003.

[23] J. S. Rosenthal, "Convergence rates of Markov chains," *SIAM Review*, vol. 37, no. 3, pp. 387–405, 1995.

[24] S. Karandikar, S. Kalyanaraman, P. Bagal, and B. Packer, "TCP rate control," *ACM Comp. Commun. Rev.*, vol. 30, no. 1, pp. 45–58, 2000.

[25] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Explicit window adaptation: a method to enhance TCP performance," *IEEE/ACM Trans. Networking*, vol. 10, no. 3, pp. 338–350, 2002.

[26] M. C. Chan and R. Ramjee, "Improving TCP/IP performance over third generation wireless networks," in *Proc. IEEE Infocom'04*, vol. 3, pp. 1893–1904.

[27] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[28] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Ecperience*, vol. 3, no. 3, pp. 115–156, Sept. 1992.

[29] The PingER project, 2004. Available at `http://www-iepm.slac.stanford.edu/pinger/`.

[30] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM Comp. Commun. Rev.*, vol. 27, no. 5, pp. 19–43, 1997.

[31] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: a true end-to-end enhancement mechanism for mobile environments," in *Proc. IEEE Infocom'00*, pp. 1537–1545.

Lin Cai (S'00-M'06) received the M.A.Sc. and Ph.D. degrees (with Outstanding Achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since July 2005, she has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Victoria, British Columbia, Canada. Her research interests span several areas in wireless communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic over wireless, mobile, ad hoc, and sensor networks. She serves as the Associate Editor for *EURASIP Journal on Wireless Communications and Networking*, and the *International Journal of Sensor Networks (IJSNet)*.

Xuemin (Sherman) Shen (M'97-SM'02) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. From September 1990 to September 1993, he was first with the Howard University, Washington D.C., and then the University of Alberta, Edmonton (Canada). Since October 1993, he has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, where he is a Professor. Dr. Shen's research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications systems, wireless security, and ad hoc and sensor networks. He is a coauthor of two books, an editor of 10 journal Special issues, and has published more than 150 papers in wireless communications and networks, control and filtering.

Dr. Shen was the Technical Co-Chair for IEEE Globecom'03 Symposium on Next Generation Networks and Internet, and ISPAN'04. He serves as the Associate Editor for *IEEE Transactions on Wireless Communications*; *IEEE Transactions on Vehicular Technology*; *ACM Wireless Networks*; *Computer Networks*; *Dynamics of Continuous, Discrete and Impulsive - Series B: Applications and Algorithms*; *Wireless Communications and Mobile Computing* (Wiley); and *International Journal Computer and Applications*. He also serves as Guest Editor for *IEEE Journal on Selected Areas in Communications*, *IEEE Wireless Communications*, and *IEEE Communications Magazine*. Dr. Shen received the Premier's Research Excellence Award (PREA) from the Province of Ontario, Canada 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. Dr. Shen is a senior member of the IEEE, and a registered Professional Engineer of Ontario, Canada.

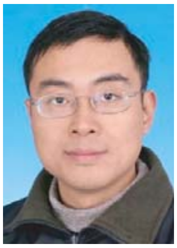
Jon W. Mark (M'62-SM'80-F'88-LF'03) received the B.A.Sc. degree from the University of Toronto in 1962, and the M.Eng. and Ph.D. degrees from McMaster University in 1968 and 1970, respectively, all in electrical engineering.

From 1962 to 1970, he was an engineer and then a senior engineer at Canadian Westinghouse Co. Ltd., Hamilton, Ontario, Canada. In September 1970 he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, where he is currently a Distinguished Professor Emeritus. He served as the Department Chairman during the period July 1984-June 1990. In 1996 he established the Centre for Wireless Communications (CWC) at the University of Waterloo and is currently serving as its founding Director.

Dr. Mark had been on sabbatical leave at the following places: IBM Thomas J. Watson Research Center, Yorktown Heights, NY, as a Visiting Research Scientist (1976-77); AT&T Bell Laboratories, Murray Hill, NJ, as a Resident Consultant (1982-83): Laboratoire MASI, Universite Pierre et Marie Curie, Paris France, as an Invited Professor (1990-91); and the Department of Electrical Engineering, National University of Singapore, as a Visiting Professor (1994-95).

He has previously worked in the areas of adaptive equalization, image and video coding, spread spectrum communications, computer communication networks, ATM switch design and traffic management. His current research interests are in broadband wireless communications, resource and mobility management, and cross domain interworking. He recently co-authored the text entitled *Wireless Communications and Networking* (Prentice-Hall 2003).

Dr. Mark is a Life Fellow of IEEE. He is the recipient of the 2000 Canadian Award for Telecommunications Research and the 2000 Award of Merit of the Education Foundation of the Federation of Chinese Canadian Professionals, 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).



Jianping Pan (S'96-M'99) received his BS and PhD degrees in computer science from Southeast University, Nanjing, China in 1994 and 1998, respectively. From 1999 to 2001, he was a postdoctoral fellow and then a research associate at the University of Waterloo, Ontario, Canada; from 2001 to 2005, he was a member of research staff at Fujitsu Labs and a research scientist at NTT MCL in Silicon Valley, California, USA. Since 2005, he has been an Assistant Professor of computer science at the University of Victoria, British Columbia, Canada. His research interests include protocols and applications for high-speed, multimedia, and mobile networks. He is a member of ACM and IEEE.