

# Analytical Framework for Simultaneous MAC Packet Transmission (SMPT) in a Multicode CDMA Wireless System

Manjunath Krishnam, *Student Member, IEEE*, Martin Reisslein, *Senior Member, IEEE*, and Frank Fitzek, *Member, IEEE*

**Abstract**—Stabilizing the throughput over wireless links is one of the key challenges in providing high-quality wireless multimedia services. Wireless links are typically stabilized by a combination of link-layer automatic repeat request (ARQ) mechanisms in conjunction with forward error correction and other physical layer techniques. In this paper, we focus on the ARQ component and study a novel class of ARQ mechanisms, referred to as simultaneous MAC packet transmission (SMPT). In contrast to the conventional ARQ mechanisms that transmit one packet at a time over the wireless air interface, SMPT exploits the parallel code channels provided by multicode code-division multiple access. SMPT stabilizes the wireless link by transmitting multiple packets in parallel in response to packet drops due to wireless link errors. While these parallel packet transmissions stabilize the link layer throughput, they also increase the interference level in a given cell of a cellular network or cluster of an *ad hoc* network. This increased interference reduces the number of traffic flows that can be simultaneously supported in a cell/cluster. We develop an analytical framework for the class of SMPT mechanisms and analyze the link-layer buffer occupancy and the code usage in a wireless system running some form of SMPT. Our analysis quantifies the tradeoff between increased link-layer quality of service and reduced number of supported flows in SMPT with good accuracy, as verified by simulations. In a typical scenario, SMPT reduces the probability of link-layer buffer overflow by over two orders of magnitude (thus enabling high-quality multimedia services, such as real-time video streaming) while supporting roughly 20% fewer flows than conventional ARQ. Our analytical framework provides a basis for resource management in wireless systems running some form of SMPT and optimizing SMPT mechanisms.

**Index Terms**—Automatic repeat request (ARQ), buffer occupancy, capacity, code-division multiple access (CDMA), link-layer quality of service (QoS), multicode, packet-loss probability, throughput.

## I. INTRODUCTION

THE fluctuation of the throughput over wireless links due to the random wireless link errors is one of the key obstacles to providing high-quality multimedia services over wire-

less links. Typically, wireless systems employ a combination of automatic repeat request (ARQ) mechanisms and forward error correction in conjunction with adaptive coding/modulation and power control to stabilize the wireless links. In this paper, we focus on the ARQ component that operates at the radio-link (MAC) layer. ARQ retransmits the packets that are dropped due to excessive bit errors on the wireless link, which could not be remedied by the forward error correction (FEC) and physical layer techniques. The ARQ mechanisms that are employed in wireless systems are typically based on one of three classical ARQ types (send-and-wait, go-back-N, and selective repeat) or a variation thereof. The common characteristic of these ARQ protocols is that they are designed for the sequential transmission of packets, i.e., the sender transmits *one packet after the other* over its radio front end onto the wireless link. (Note that the packets are not necessarily transmitted in sequence, e.g., an “older” packet may be retransmitted after a “newer” packet. What we mean by sequential transmission is that the sender transmits at any given time at most one packet.) Many modern wireless systems are based on code-division multiple access (CDMA). In addition, many of these systems, such as IS-95 (Rev B) [1] and Universal Mobile Telecommunications System (UMTS) [2], are *multicode* CDMA systems, i.e., they allow the sender to transmit multiple packets simultaneously (in parallel) to a given receiver by using multiple CDMA code channels. The classical ARQ mechanisms, however, are designed to transmit one packet after the other and thus do not take advantage of the multiple parallel CDMA code channels. In this paper, we study a novel class of ARQ mechanisms for multicode CDMA wireless systems. We refer to the novel class of ARQ mechanisms as *simultaneous MAC packet transmission (SMPT)* mechanisms.

In contrast to the classical ARQ mechanisms, which were designed for a single channel, SMPT exploits the parallel code channels provided by multicode CDMA. SMPT transmits multiple packets in parallel on multiple CDMA codes to overcome the packet drops on the unreliable wireless link. By transmitting multiple packets on parallel CDMA codes in response to packet drops, SMPT stabilizes the link-layer throughput and thus provides a basis for high-quality multimedia services over wireless links. Indeed, our preliminary simulation results [3], [4] indicate that SMPT efficiently supports the real-time streaming of high-quality video over wireless links. Simulations, however, provide only limited insights into the behavior of SMPT. Therefore, in this paper, we develop an analytical framework for

Manuscript received January 23, 2003; revised September 3, 2003 and November 7, 2003. This work was supported in part by the National Science Foundation through Grant CAREER ANI-0133252 and the State of Arizona through the IT301 initiative. An overview of this work was presented at the High-Speed Networking Workshop, San Francisco, CA, March 2003.

M. Krishnam and M. Reisslein are with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-5706 USA (e-mail: manjunath@asu.edu; reisslein@asu.edu).

F. Fitzek was with Acticom GmbH, Berlin 13507, Germany. He is now with the Department of Communication Technology, University of Aalborg, Aalborg, Denmark (e-mail: ff@kom.auc.dk).

Digital Object Identifier 10.1109/TVT.2003.822329

SMPT that provides a sound theoretical basis for resource management and optimization of SMPT.

SMPT operates exclusively at the link layer and does not require any information from the other protocol layers. Thus, SMPT preserves the isolation of the network protocol layers, allowing for easy deployment. (Information from the other layers, e.g., SIR measurements from the physical layer or time stamps of the application layer or the application layer traffic may be exploited in refined cross-layer designs, which are beyond the scope of this paper.) SMPT does not require any coordination among the transmitting wireless terminals, i.e., there is no centralized controller or scheduler required. Instead, the SMPT mechanism in each client monitors whether its packet transmissions are successful or unsuccessful (which depends largely on the interference level) and reacts as detailed in Section III. SMPT is, therefore, especially well suited for *ad hoc* wireless networks where distributed uncoordinated wireless terminals communicate with each other and share the common interference environment of a local cluster. It is also well suited for low-overhead uplink transmissions in a cellular environment, where several uncoordinated wireless terminals transmit to a central base station.

By transmitting multiple packets on parallel CDMA codes, SMPT stabilizes the link-layer throughput, i.e., strives to avoid excessive buffer buildup at the link layer, at the expense of increased interference. [We assume throughout that the used CDMA codes are correlated pseudonoise (PN) codes; thus, using more codes increases the interference.] This increased interference may cause more packet drops on the wireless links, which in turn call for the use of more codes. Clearly, this will lead to instabilities in the form of excessive buffer buildup (and buffer overflow) if the traffic volume exceeds a critical threshold, which is typically referred to as *capacity*. In this paper, we develop an analytical framework to quantify the tradeoff between the increased link-layer quality of service (QoS) and the reduced capacity when running some form of SMPT in a multicode CDMA system.

This paper is organized as follows. In the following section, we review related work. In Section II, we give an overview of the considered multicode CDMA system and describe the considered traffic model and performance metrics, which are throughput and probability of overflow of the link-layer buffer. We also describe the two considered wireless link models: the independent link model and the interference link model. Both models are based on an underlying two-state (good and bad) Markov model. In the independent link model, packets are dropped in both states with fixed probabilities. The interference link model, on the other hand, captures the interference in the cluster/cell by making the packet-drop probability in the good state a function of the interference level. In Section III, we introduce SMPT and describe some of its forms.

Our analytical framework for SMPT consists of three main components, which are presented in Sections IV–VI. In the first component, we derive the distribution of the link-layer buffer occupancy of an individual client with the independent link model. Based on the link buffer occupancy distribution, we calculate the channel usage of an individual client as well as the channel usage (interference level) in the cluster/cell in the

second component (in Section V). Both of these channel-usage calculations rely on the independent link model. In Section VI, we incorporate the interactions between the ongoing transmissions in the cluster/cell (i.e., the effect of the interference) into the analysis. Toward this end, we combine the buffer analysis of an individual client (Section IV) and the analysis of the interference level (Section V) with the interference link model (Section II-C). In Section VII, we study the SMPT performance for bursty traffic. Finally, we summarize our conclusions in Section VIII.

#### A. Related Work

A large body of work has studied the capacity (maximum number of supported client) in wireless systems subject to physical layer QoS requirements, such as a minimum threshold for the energy-per-bit-to-interference ratio (see, for instance, [5]–[8]). In contrast, in this paper, we study the performance of wireless systems at the radio-link medium access control (MAC) layer. There exists an extensive literature on the link-layer performance of wireless systems. The classical ARQ mechanisms for a *single channel* have been analyzed thoroughly (see, for instance, [9]–[13]). Also, the link-layer performance of forward error correction (FEC)-based schemes have been studied (see, for instance, [14] and [15]).

Recently, hybrid ARQ schemes that combine some form of FEC with ARQ packet retransmissions have received considerable interest, see e.g., [16]–[18]. Many of these hybrid ARQ schemes adapt to channel variations, e.g., by adjusting the FEC coding rate, packet length, etc. (see, for instance, [19]–[23]). These hybrid ARQ studies are largely orthogonal to our study on SMPT. We envision that SMPT's packet transmissions over the multiple parallel CDMA code channels may be combined with FEC to form hybrid SMPT schemes. We also envision that channel-adaptive mechanisms (in addition to SMPT's underlying channel-adaptive packet transmission/retransmission) may be added to those hybrid SMPT schemes. We leave these directions to be explored in future work.

Recently, ARQ mechanisms that are specifically designed for the transmission of video over wireless links have been developed. A hybrid ARQ scheme for video transmission over an 802.11-based wireless local area network (LAN) is developed in [24]. A hybrid ARQ scheme for video transmission over a single wireless channel is studied in [25]. An adaptive video encoder is embedded in the sender packet transmission-receive acknowledgment feedback loop in [26]; the encoder dynamically adjusts the encoding rate based on acknowledgment from the receiver. An interesting variation of ARQ is studied in [27] for transmitting two-layer encoded video over *ad hoc* wireless networks. In this scheme, the base and enhancement layers are transmitted over disjoint (multihop) paths. If a base layer is lost, then it is retransmitted over the enhancement layer path.

More closely related to our work are the studies on providing link-layer QoS in multicode CDMA systems. A conceptual framework for a multicode CDMA-based wireless multimedia network is described in [28]. This work points to the possibility of conducting ARQ retransmissions over a separate CDMA code channel but does not analyze this approach.

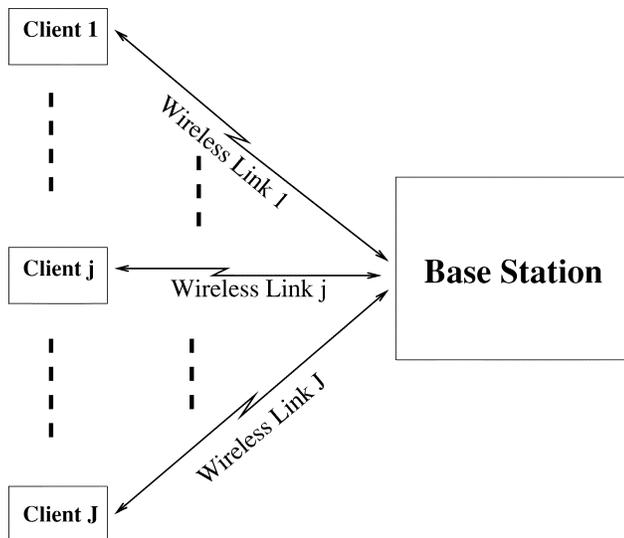


Fig. 1. System architecture:  $J$  wireless clients conduct uncoordinated uplink transmissions to a base station.

Admission-control strategies for multiple traffic classes (each requiring a different, but fixed, number of code channels) in a multicode CDMA system are studied in [29]. A hybrid ARQ scheme for transmitting video in a multicode CDMA system is developed and analyzed in [30]. In this scheme, the number of code channels used by a given client is constant. If packets are lost, the scheme reduces the FEC and, thus, increases the transmission (bit) rate for payload data to accommodate retransmissions on the fixed number of CDMA code channels. Video transmission in multicode CDMA systems is also studied in [31]. The scheme proposed in [31] is similar to ours in that multiple codes are used in parallel on a dynamic basis. The main difference between [31] and SMPT is that [31] requires a significant amount of coordination among the videos being transmitted. For instance, the video streams are aligned such that a (typically large) Intracoded (I)-frame of one video stream does not coincide with the I-frame of another video stream. SMPT, on the other hand, does not require any coordination among the ongoing traffic flows and is, thus, well suited for wireless networks with little or no coordination among the wireless terminals, such as *ad hoc* networks.

We finally note that a number of schemes have been developed for providing link-layer QoS in wireless multicode CDMA systems with a fixed infrastructure and a *central* base station (see, for instance, [32] and [33], which are based on the DQRUMA [34], or the LIDA/BALI approach [35], [36]). In contrast, SMPT mechanisms are distributed, i.e., SMPT does not require a central unit for packet scheduling and is, thus, well suited for *ad hoc* wireless networks.

## II. OVERVIEW OF MULTICODE CDMA SYSTEM

The studied SMPT packet-transmission mechanisms do not require any coordination between the transmitting terminals and, thus, can be deployed in cellular networks as well as *ad hoc* networks. However, to fix ideas for our discussion, we consider the uplink communication in a cellular system, as illustrated in Fig. 1. Let  $J$  denote the number of wireless (and possibly

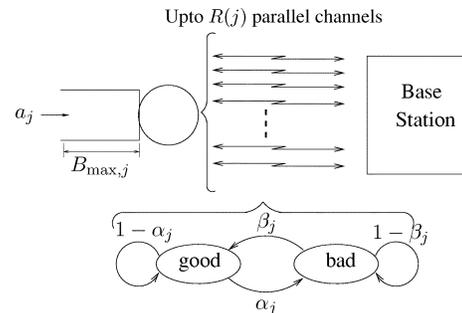


Fig. 2. Client buffer of capacity  $B_{max,j}$  packets is served by up to  $R(j)$  parallel code channels. The wireless link (consisting of up to  $R(j)$  channels) is modeled with a two-state Markov chain.

mobile) terminals transmitting to the base station. We consider a multicode CDMA system where the base station allocates to each wireless terminal a set of unique PN code sequences for uplink transmission. Let  $R(j)$ ,  $j = 1, \dots, J$  denote the number of parallel code channels supported by the radio front end of terminal  $j$  and suppose that the base station has allocated the terminal at least  $R(j)$  sequences. Throughout we assume perfect power control, ensuring that each wireless terminal is received at the base station with the same power level (which is typical for modern wireless systems [37]). We consider a system with a time division duplex (TDD) timing structure. Specifically, time is divided into fixed-length *slots*. Each slot is subdivided into a fixed-length *uplink subslot* followed by a *downlink subslot* of fixed length. The uplink subslot is used for transmissions in the uplink (reverse), i.e., wireless terminal to base station, direction. The downlink subslot is used for transmissions in the downlink (forward), i.e., base station to wireless terminal, direction.

We assume that the wireless terminals transmit fixed-size packets (link-layer protocol data units) to the base station. The packet size is set such that one CDMA code channel accommodates exactly one packet in one uplink subslot. Note that by using its  $R(j)$  parallel code channels, terminal  $j$  can send up to  $R(j)$  packets in an uplink subslot.

### A. Client Model

We initially assume that client  $j$  generates a new packet independently with probability  $a_j$ ,  $0 < a_j \leq 1$  at the beginning of an uplink subslot. We refer to  $a_j$  as the *activity factor* of client  $j$ . (We initially consider this nonbursty Bernoulli traffic-generation process to keep the system analysis relatively simple and to highlight the main features of our analytical framework; bursty traffic arrivals are considered in Section VII. Furthermore, in this paper we focus on a scenario in which clients generate at most one packet per slot. Multirate traffic scenarios in which some high-speed clients may generate multiple packets per slot are considered in future work.) Client  $j$  has a buffer of capacity  $B_{max,j}$  packets, as illustrated in Fig. 2. The newly generated packet is placed in the buffer (provided there is free buffer capacity). Packets are transmitted in the uplink subslot out of the client's buffer to the base station according to the SMPT mechanisms described in detail in Section III. During the downlink subslot, the base station acknowledges the packet correctly received in the preceding uplink subslot. (Typical hard-

ware configurations of modern wireless communication systems allow the base station to acknowledge the packets received in an uplink subslot immediately in the following downlink subslot [38].) Each acknowledged packet is immediately flushed from the client's buffer. A packet that was transmitted in an uplink slot, but is not acknowledged in the subsequent downlink slot, stays in the buffer. The client will attempt to retransmit the unacknowledged packet(s) in the following uplink subslot(s), according to some form of SMPT as outlined in Section III. We choose the simple first-come-first-served service discipline with tail drop to fix ideas for the development of our fundamental analytical framework. A wide variety of other service disciplines may be considered, e.g., service disciplines that take packet time stamps into consideration or service disciplines that drop packets that have exceeded a prespecified delay bound (see e.g., [39]).

Let  $B_j$ ,  $j = 1, \dots, J$  be a discrete random variable denoting the number of packets in the buffer of client  $j$  at the end of the downlink subslot [after the acknowledged packet(s) have been flushed from the buffer] in steady state. Note that  $P(B_j = b_j)$ ,  $b_j = 0, \dots, B_{\max,j}$  denotes the steady-state probability of client buffer  $j$  holding  $b_j$  packets at the end of the downlink subslot. Also, note that a new packet arriving to a full buffer ( $b_j = B_{\max,j}$ ) is lost (tail drop). We define the packet-loss probability  $P_l(j)$  for client  $j$  as the probability that a given newly generated packet finds the buffer full, i.e.,  $P_l(j) = P(B_j = B_{\max,j})$ . We define the average loss probability among the clients in the cell as

$$P_l = \frac{1}{J} \sum_{j=1}^J P_l(j). \quad (1)$$

We define the throughput of client  $j$   $\text{TH}(j)$  as the long-run average rate at which the packets generated by client  $j$  are (successfully) transmitted, i.e.,

$$\text{TH}(j) = a_j \cdot [1 - P_l(j)] \quad (2)$$

in packets per slot. We define the aggregate throughput  $\text{TH}$  as the long-run average rate at which packets are successfully transmitted from the  $J$  clients in the cell to the base station, i.e.,

$$\text{TH} = \sum_{j=1}^J \text{TH}(j) \quad (3)$$

in packets per slot.

### B. Wireless Link Model

We employ the widely used two-state Markov Chain model [40], [41] (also referred to as the Gilbert–Elliot model) as our basic underlying wireless link model. This two-state Markov Chain model captures the correlated errors that are typical for wireless links. We model the wireless link (consisting of up to  $R(j)$  parallel code channels) between each wireless terminal and the base station as an independent discrete time Markov Chain with two states: “good” and “bad.” In the good state, packet transmissions are generally successful, but some packets may be unsuccessful with a probability that may depend upon

the interference level (see Section II-C for details). The bad state corresponds to a deep fade (or shadowing) in which all packet transmissions are unsuccessful. This two-state model has been found to be a useful and accurate model for link-layer (packet level) analysis [13], [42]–[46]. The two-state model may be obtained from more complex wireless channel models, which may incorporate adaptive error-control techniques, using weak lumpability or stochastic bounding techniques [47]. While we model the  $J$  wireless links in the cell as  $J$  independent Markov Chains, we do introduce dependencies between the links when modeling the link errors in the good state. These dependencies capture the interference between the ongoing transmissions in the cell (see Section II-C for details).

In our channel model, state transitions occur at the end of each downlink subslot. We refer to a slot (consisting of an uplink and downlink subslot) during which wireless link  $j$  is in the good state as a *good slot* for link  $j$ . We refer to a slot during which wireless link  $j$  is in the bad state as a *bad slot* for link  $j$ . Throughout, we assume that all parallel code channels of a given wireless link (between a particular client and the base station) experience either a good slot or a bad slot.

1) *Model Parameters:* We denote  $\alpha_j$  for the probability that a transition takes link  $j$  from the good state to the bad state, given that link  $j$  is currently in the good state (with the complementary probability  $1 - \alpha_j$ , the link stays in the good state). We denote  $\beta_j$  for link  $j$ 's transition probability from the bad state to the good state. We denote  $P_{\text{good}}^j$  ( $P_{\text{bad}}^j$ ) for the steady-state probability that link  $j$  is in the good (bad) state in a given slot, i.e.,  $P_{\text{good}}^j = \beta_j / (\alpha_j + \beta_j)$  and  $P_{\text{bad}}^j = \alpha_j / (\alpha_j + \beta_j)$ . We denote  $\bar{T}_{\text{good}}^j$  ( $\bar{T}_{\text{bad}}^j$ ) for link  $j$ 's average sojourn time in the good (bad) state in slots, i.e.,  $\bar{T}_{\text{good}}^j$  ( $\bar{T}_{\text{bad}}^j$ ) is the average number of consecutive good (bad) slots of link  $j$ . Clearly,  $\bar{T}_{\text{good}}^j = 1/\alpha_j$  slots and  $\bar{T}_{\text{bad}}^j = 1/\beta_j$  slots. We note that for a flat-fading channel, the Markov Chain parameters may be derived in terms of the Rayleigh-fading parameters [48]. The steady-state probabilities are given in terms of the ratio of the Rayleigh-fading envelope to the local root mean square level by  $P_{\text{good}}^j = e^{-\rho_j^2}$  (and  $P_{\text{bad}}^j = 1 - e^{-\rho_j^2}$ ). Typical values of the fade margin at the radio front end of the wireless terminal are between 5 and 20 dB (i.e.,  $\rho_j$  is typically between  $-20$  and  $-5$  dB). This corresponds to typical values between 0.9 and 0.999 for  $P_{\text{good}}^j$  and values between 0.001 and 0.1 for  $P_{\text{bad}}^j$ . We conservatively consider  $P_{\text{good}}^j = 0.9$  and  $P_{\text{bad}}^j = 0.1$  in our numerical work in this paper. The average sojourn time in the bad state is given by  $(e^{\rho_j^2} - 1)/(2\pi\rho_j f_j)$ , where  $f_j$  denotes the maximum Doppler frequency given by  $f_j = v_j/\lambda$ , with  $v_j$  denoting the speed of terminal  $j$  and  $\lambda$  denoting the carrier wavelength. The UMTS system carrier wavelength of  $\lambda = 0.159$  m (corresponding to a carrier frequency of 1.855 GHz) and a typical mobile speed of  $v_j = 0.2$  m/s suggest  $\bar{T}_{\text{bad}}^j = 32$  ms. For our numerical work in this paper, we consider a slot length of 10 ms and  $\beta = 1/3$ .

2) *Period:* As groundwork for our analytical framework, we analyze the lengths of the runs of consecutive good and bad slots (i.e., the sojourn times) in the Markov Chain model of wireless link  $j$  in some more detail. We define a *period* as a run of consecutive bad slots followed by a run of consecutive good slots. Let

$T_{\text{good}}^j$  ( $T_{\text{bad}}^j$ ) be a discrete random variable denoting the number of consecutive good slots (bad slots) in a given period of link  $j$ . Clearly

$$P(T_{\text{bad}}^j = m) = \begin{cases} \beta_j(1 - \beta_j)^{m-1}, & \forall m > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

$$P(T_{\text{good}}^j = n) = \begin{cases} \alpha_j(1 - \alpha_j)^{n-1}, & \forall n > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Note that, by the Markovian property,  $T_{\text{bad}}^j$  and  $T_{\text{good}}^j$  are independent random variables, i.e., the number of consecutive bad slots in a given period is independent of the number of consecutive good slots in that same period. Hence, the probability that a given period consists of  $m$  consecutive bad slots followed by  $n$  consecutive good slots, which we denote by  $\pi(m, n)$ , is given by

$$\pi(m, n) = P(T_{\text{bad}}^j = m, T_{\text{good}}^j = n) \quad (6)$$

$$= \begin{cases} \alpha_j \beta_j (1 - \alpha_j)^{n-1} (1 - \beta_j)^{m-1}, & \forall m, n > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Finally, let  $\bar{T}_j$  denote the average length of a period in slots and note that  $\bar{T}_j = E[T_{\text{bad}}^j] + E[T_{\text{good}}^j] = 1/\beta_j + 1/\alpha_j$ .

### C. Packet-Drop Probability in Good/Bad State

Throughout, we set the packet-drop probability in the bad state to  $q_{\text{bad}}^j = 1$ , i.e., if link  $j$  is in the bad state, all packets sent between client  $j$  and the base station are dropped on the wireless link with probability one. We consider two approaches for modeling the packet-drop probability in the good state. In the first approach, each packet is independently dropped in the good state with a fixed probability  $q_{\text{good}}^j$ . Note that, in this first approach, the model of a given wireless link is completely independent from the models of the other wireless links in the cell. In other words, with this link model a given client does not “feel” the transmission activities of the other clients in the cell. Therefore, we refer to this model as the *independent link model*.

In the second approach, the packet-drop probability in the good state is a function of the interference level, i.e., the total number of codes used by the other clients in the cell. Let  $i$ ,  $i = 0, \dots, (J-1) \cdot R$  denote the total number of currently interfering PN CDMA codes in the cell. (We assume here that the codes of a given client are orthogonal, achieved, for instance, by subcode concatenation, such that there is no self-interference.) We employ the widely used Holtzman approximation [49] to calculate the bit-error probability  $q_{\text{bit}}(i)$  resulting from an interference level of  $i$  codes

$$q_{\text{bit}}(i) = \frac{2}{3}Q \left[ \sqrt{\frac{3G}{i-1}} \right] + \frac{1}{6}Q \left[ \frac{G}{\sqrt{\frac{(i-1)i}{3}} + \sqrt{3}\sigma} \right] + \frac{1}{6}Q \left[ \frac{G}{\sqrt{\frac{(i-1)i}{3}} - \sqrt{3}\sigma} \right] \quad (8)$$

where

$$\sigma^2 = (i-1) \left[ G^2 \frac{23}{360} + (G-1) \left( \frac{1}{20} + \frac{i-2}{36} \right) \right] \quad (9)$$

and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt, \quad x \geq 0 \quad (10)$$

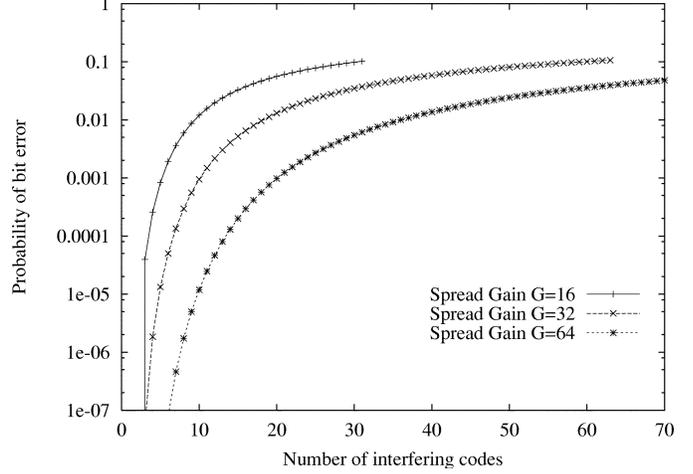


Fig. 3. Bit-error probability  $q_{\text{bit}}(i)$  as a function of total number of interfering codes  $i$  for different spreading gains  $G$ .

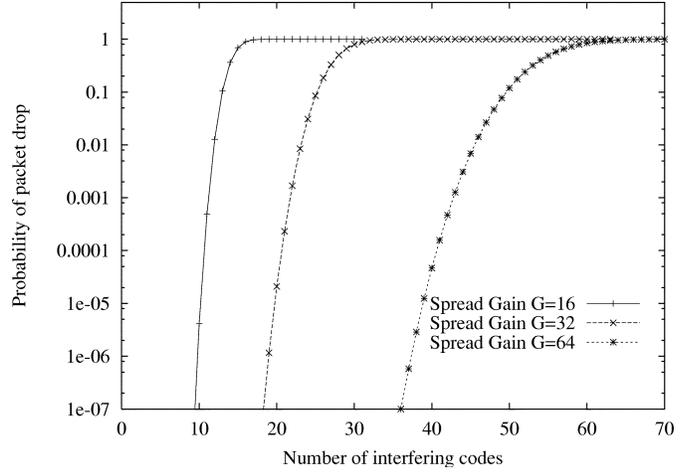


Fig. 4. Packet-drop probability  $q_{\text{good}}(i)$  as a function of total number of interfering codes  $i$  for different spreading gains  $G$ .

is the complementary error function and  $G$  denotes the spreading gain. The Holtzman approximation calculates the bit-error rate caused by the multiple-access interference (neglecting the effects of thermal noise) for a system with equal received signal powers and randomly interfering signature sequences. Based on the bit-error probability  $q_{\text{bit}}(i)$ , we calculate the packet-drop probability in the good state  $q_{\text{good}}(i)$  by considering a simple static FEC as follows. We set the packet length to 1023 b and employ static forward error correction that can correct upto 30 bit errors. (We use these settings to fix ideas; our analytical approach is valid for arbitrary settings of these parameters.) Thus

$$q_{\text{good}}(i) = \sum_{e=0}^{30} \binom{1023}{e} [q_{\text{bit}}(i)]^e \cdot [1 - q_{\text{bit}}(i)]^{1023-e}. \quad (11)$$

Note that this second approach captures the interference effect of the ongoing uplink transmissions in a cell in the models for the individual wireless links, i.e., a given client “feels” the transmissions of the other clients in the cell. We refer to this model as the *interference link model*. Figs. 3 and 4 depict the bit-error probability  $q_{\text{bit}}$  and the packet-drop probability  $q_{\text{good}}$  as a function of the total number of interfering codes  $i$  for different spreading gains  $G$ . We emphasize that we use the Holtzman approximation for the bit-error probability and the static FEC for the packet-

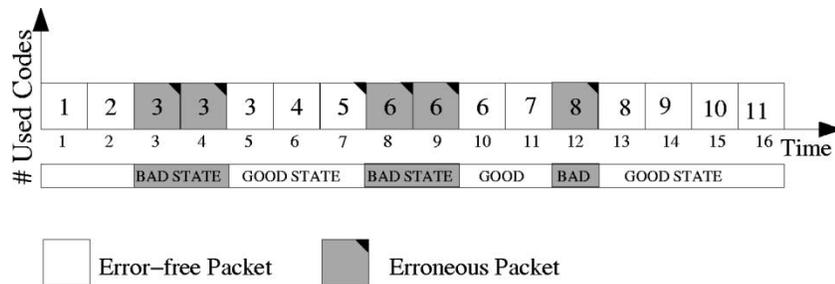


Fig. 5. Conventional ARQ. Packet drops on wireless link result in throughput fluctuations and buffer buildup at the link layer.

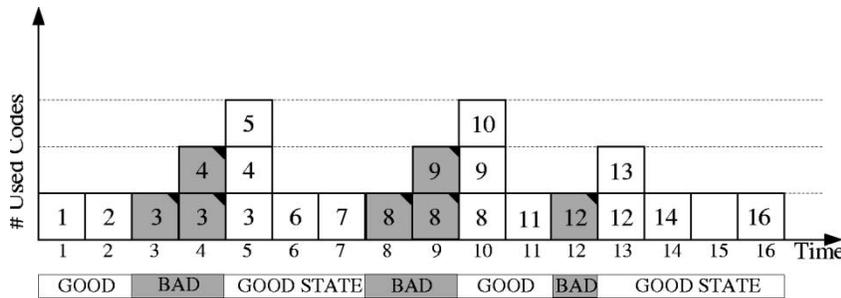


Fig. 6. Basic SMPT strives to stabilize throughput over wireless link and avoid buffer buildup.

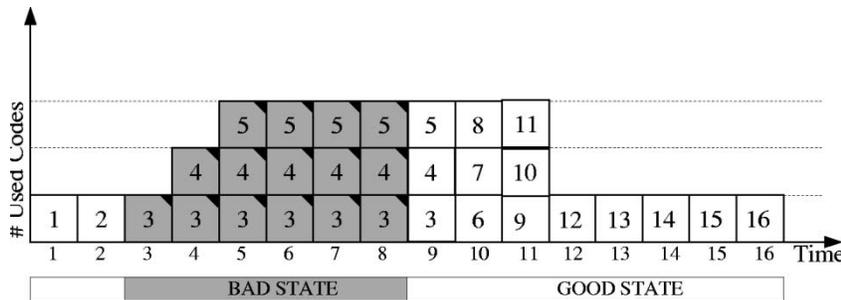


Fig. 7. Inefficiency of basic SMPT in a scenario where link errors are correlated.

error probability only to fix ideas and to establish a baseline reference for our comparison of SMPT with conventional ARQ mechanisms in Section VI-A. Our analytical framework only assumes that there is some way to obtain the packet-drop probability  $q_{\text{good}}$  as a function of the number of interfering codes  $i$ .

### III. OVERVIEW OF SMPT

In this section, we introduce SMPT, a novel class of ARQ mechanisms for multicode CDMA systems. First, we recall that in the considered setting where a packet sent in an uplink subslot is immediately acknowledged in the following downlink subslot, all the conventional ARQ mechanisms (send-and-wait, go-back-N, and selective repeat) work in send-and-wait fashion. When a packet is dropped on the wireless link, the client retransmits the packet until it is successfully transmitted (and acknowledged), as illustrated in Fig. 5. Clearly, with this approach, packet drops on the wireless link delay the transmission of the subsequent packets and thus lead to throughput fluctuations and buffer buildup at the link layer in the client. This buffer buildup increases the probability of losing a newly generated packet due to buffer overflow, which we analyze in this paper. The buffer buildup also increases the packet delay and packet jitter; the analytical study of these metrics is beyond the scope of this paper and is a topic of future work.

The SMPT mechanisms strive to stabilize the wireless link by transmitting multiple packets in parallel using multiple CDMA codes (one for each packet) when a packet is dropped on the wireless link. Suppose that a transmitted packet is not successfully acknowledged. With *basic SMPT*, in the next uplink subslot, the client transmits the lost packet *and* the subsequent packet (which would have been transmitted in that subslot, had there not been a packet drop) on two CDMA codes, as illustrated in Fig. 6. If these packets are successfully received and acknowledged, the client returns to sending one packet using one CDMA code in the next uplink subslot. Otherwise (i.e., if the packets are not successfully acknowledged), the client sends three packets (the two unsuccessful packets plus the packet next in line) using three CDMA codes. This process continues until the packets are successful or the client has “ramped up” to using a prespecified maximum number  $R$  of CDMA codes.

The outlined basic SMPT mechanism performs well when the packet drops (i.e., bad slots) on the wireless link are independently distributed. However, for the typically correlated bad slots in real wireless systems, the basic SMPT mechanism uses the CDMA codes inefficiently, as illustrated in Fig. 7. The client’s parallel transmissions in the bad slots increase the interference level in the cell without reducing the backlog in the client. To address this shortcoming, forms of SMPT that incor-

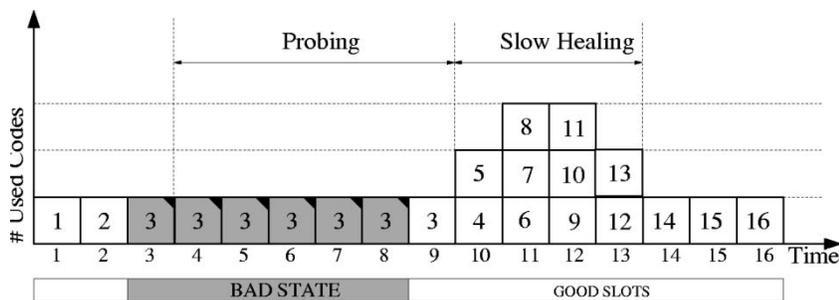


Fig. 8. Slow-healing SMPT.

porate *link probing* are introduced. If a transmitted packet is not acknowledged, the client retransmits the lost packet (as a link probe) using only one single CDMA code until this packet is successfully acknowledged. (We note that, in cross-layer designs, the link may be probed using the physical layer infrastructure, e.g., by reading SIR measurements, instead of sending a probing packet. We consider link-layer probing with probing packets in this paper to find the fundamental performance characteristics of SMPT with respect to conventional ARQ mechanisms on the link layer.) The client then clears the backlog that has accumulated during the probing. With the so-called *slow-healing SMPT*, the client ramps up by using two CDMA codes in the slot right after the probing packet was successfully acknowledged, three codes in the subsequent slot, and so on, until all  $R$  codes are used, as illustrated in Fig. 8. The client returns to probing if all packets sent in parallel in an uplink subslot are dropped on the wireless link. [Note that this can be a result of either a bad slot or the independent packet drops with probability  $q_{\text{good}}(i)$  in a good slot]. If only a subset of the packets sent in an uplink subslot are dropped, then the client does not start probing.

We note that the outlined forms of SMPT are only examples and that many other forms of SMPT are possible. For instance, one variation of slow-healing SMPT is to ramp up to two codes with a prespecified probability after the successful probing packet is received, then ramp up to three codes with a prespecified probability, and so on. Indeed, there is a wide-open design space for forms of SMPT and for finding the optimal form for a given setting. The exploration of this design space and the optimization of the form of SMPT is beyond the scope of this paper. In this paper, we focus on establishing an analytical framework that can accommodate any possible form of SMPT and, thus, provide a basis for further exploration and optimization. Throughout this paper, when illustrating specific aspects of our analytical framework, we consider basic SMPT and slow-healing SMPT. Basic SMPT (which does not include link probing) is considered because it gives very simple analytical expressions. Slow-healing SMPT is considered as an illustrative example of a form of SMPT with link probing, which is important for the typically bursty error patterns of wireless systems.

#### IV. ANALYSIS OF BUFFER OCCUPANCY OF WIRELESS CLIENT WITH INDEPENDENT LINK MODEL

In this section, we analyze the buffer occupancy of a wireless client  $j$  that uses SMPT to transmit data to the base sta-

tion, as illustrated in Fig. 2. The analysis in this section relies on the independent link model, i.e., the considered client  $j$  is not affected by the transmissions of the other clients in the cell. (The interference link model is considered in Section VI.) Our goal is to evaluate the steady-state probability  $P(B_j = b_j)$ ,  $b_j = 0, \dots, B_{\text{max},j}$  that client  $j$  holds  $b_j$  packets at the end of a downlink subslot. In order to not obscure the main idea of our approach, we initially restrict the client's activity factor to  $a_j = 1$ . Subsequently, we will extend the analysis to  $a_j < 1$ . Also, we consider initially an infinite buffer capacity, i.e.,  $B_{\text{max},j} = \infty$ . We subsequently refine the analysis to finite buffers  $B_{\text{max},j}$ . In addition, we initially assume that packets are not dropped in the good link state, i.e.,  $q_{\text{good}} = 0$  (we will relax this assumption in Section IV-C). Also, initially we focus on obtaining the steady-state probability that the client buffer  $j$  holds  $b_j$  packets at the end of the downlink subslot that is the last subslot of a period. We denote this steady-state probability by  $P(B_j^g = b_j)$ ,  $b_j = 0, \dots, B_{\text{max},j}$ . Note that  $P(B_j^g = b_j)$  is the steady-state probability that client buffer holds  $b_j$  packets at the end of a period, i.e., at the end of a run of consecutive good channel states. The main idea in the calculation of the steady-state probabilities  $P(B_j^g = b_j)$ ,  $b_j = 0, \dots, B_{\text{max},j}$  is to construct an irreducible positive-recurrent discrete-time Markov chain with the states  $B_j^g = 0, \dots, B_j^g = B_{\text{max},j}$ . The Markov chain makes state transitions at the end of each period.

Let  $\Pr\{b_n|b_o\}$ ,  $b_n, b_o = 0, \dots, B_{\text{max},j}$  denote the transition probabilities of the Markov chain, i.e.,  $\Pr\{b_n|b_o\}$  is the probability that the backlog at the end of a period is  $b_n$  packets given that the backlog at the beginning of the period (i.e., at the end of the preceding period) is  $b_o$  packets. Toward the calculation of these transition probabilities, let  $\Pr\{b_n, m|b_o\}$  denote the probability that there are  $m$ ,  $m > 0$  consecutive bad slots in the period and that there is a nonzero backlog of  $b_n$ ,  $b_n = 1, \dots, B_{\text{max},j}$  packets at the end of the period given that the backlog at the beginning of the period is  $b_o$ ,  $b_o = 0, \dots, B_{\text{max},j}$  packets. These probabilities are given in Table I for basic and slow-healing SMPT, described in Section III. [Recall here that  $\pi(m, n)$  denotes the probability that a given period has  $m$  consecutive bad slots followed by  $n$  consecutive good slots. See (7).] The  $\Pr\{b_n, m|b_o\}$  for other forms would be derived in analogous fashion and then incorporated into our overall analysis framework. We now outline how these expressions are derived. For any form of SMPT, when there are  $b_o$  backlogged packets at the beginning of the period and there are  $m$  consecutive bad slots in the period, then there are  $\max(b_o + m, B_{\text{max},j})$  backlogged packets at the end of the run of bad slots. This is because

TABLE I  
PROBABILITIES  $\Pr\{b_n, n|b_o\}$  FOR DIFFERENT FORMS OF SMPT

SMPT	$\Pr\{b_n, n b_o\}$
Basic	$\Pr\{b_n, m b_o\} = \begin{cases} \pi\left(m, \left\lfloor \frac{b_o+m-b_n}{R-1} \right\rfloor\right) & \text{if } \frac{b_o+m-b_n}{R-1} = \left\lfloor \frac{b_o+m-b_n}{R-1} \right\rfloor \\ 0 & \text{otherwise.} \end{cases}$
Slow Heal	$\Pr\{b_n, m b_o\} = \begin{cases} \pi\left(m, \left\lfloor \frac{1+\sqrt{1+8(b_o+m-b_n)}}{2} \right\rfloor\right) & \text{if } b_o+m-b_n \leq R \cdot (R-1)/2 \\ & \text{and } b_o+m-b_n = \left(\left\lfloor \frac{1+\sqrt{1+8(b_o+m-b_n)}}{2} \right\rfloor\right) \left(\left\lfloor \frac{1+\sqrt{1+8(b_o+m-b_n)}}{2} \right\rfloor - 1\right) / 2 \\ \pi\left(m, \left\lfloor \frac{b_o+m-b_n-R \cdot (R-1)/2}{R-1} \right\rfloor + R\right) & \text{if } b_o+m-b_n > R \cdot (R-1)/2 \\ & \text{and } \frac{b_o+m-b_n-R \cdot (R-1)/2}{R-1} = \left\lfloor \frac{b_o+m-b_n-R \cdot (R-1)/2}{R-1} \right\rfloor \\ 0 & \text{otherwise.} \end{cases}$

a new packet is generated in every slot with the considered activity factor  $a_j = 1$ . Now consider basic SMPT, which transmits  $R$  packets in every good slot of a period that ends with a backlog  $b_n \geq 1$ . (Recall that the expressions in Table I hold only for those periods that end with a backlog of at least one packet.) With  $R$  packets being transmitted in every good slot, the backlog is effectively reduced by  $R - 1$  packets in every good slot (because one new packet is generated in every slot with  $a_j = 1$ ). Thus, with  $R = 2$ , there are  $(b_o + m - b_n)/(R - 1)$  good slots required to reduce the backlog from  $b_o + m$  packets at the end of the run of bad slots to  $b_n$  packets at the end of the run of good slots (i.e., the end of the period). With  $R > 2$ ,  $\lfloor (b_o + m - b_n)/(R - 1) \rfloor$  good slots are required and only scenarios in which the reduced backlog  $(b_o + m - b_n)$  is an integer multiple of  $R - 1$  are feasible, resulting in the  $\Pr\{b_n, m|b_o\}$  given in Table I for basic SMPT.

With slow-healing SMPT, the backlog-clearing process has two phases, as discussed in Section III. In the first,  $R$  good slots, the number of transmitted packets is increased from one to  $R$  simultaneously transmitted packets. Since one new packet is generated in each of these slots, the backlog is effectively reduced by  $R \cdot (R - 1)/2$  packets in this “ramping-up” phase of duration  $R$  slots. In each subsequent good slot, the backlog is reduced by  $R - 1$  packets. Now, consider a period in which the cleared backlog  $b_o + m - b_n$  is less than or equal to  $R \cdot (R - 1)/2$ . Since we stay within the “ramping-up” phase in such a period, the number  $n$  of good slots required to achieve a backlog reduction of  $b_o + m - b_n$  packets is given as the integer solution of  $n(n - 1)/2 = b_o + m - b_n$ , resulting in the first expression for  $\Pr\{b_n, m|b_o\}$ , given in Table I.

Next, consider a period in which the cleared backlog  $b_o + m - b_n$  is larger than  $R \cdot (R - 1)/2$  packets. In this case,  $R \cdot (R - 1)/2$  packets of backlog are cleared during the ramping-up phase, leaving  $b_o + m - R \cdot (R - 1)/2$  packets of backlog to be cleared in the subsequent good slots. During the subsequent good slots,

the backlog-clearing process is equivalent to the basic SMPT behavior, resulting in the second expression for slow-healing SMPT in Table I.

The transition probabilities  $\Pr\{b_n|b_o\}$  with  $b_n > 0$  are then obtained as

$$\Pr\{b_n|b_o\} = \sum_{m>0} \Pr\{b_n, m|b_o\}.$$

The transition probability  $\Pr\{0|b_o\}$  is given by

$$\Pr\{0|b_o\} = 1 - \sum_{b=1}^{B_{\max,j}} \Pr\{b|b_o\}.$$

Based on the transition probabilities  $\Pr\{b_n|b_o\}$ ,  $b_o, b_n = 0, \dots, B_{\max,j}$ , we find the steady-state probabilities  $P(B_j^g = b_j)$ ,  $b_j = 0, 1, \dots, B_{\max,j}$  using standard techniques [50], [51].

#### A. Refined Analysis for Activity Factor $a_j < 1$

Note that the above analysis is for an activity factor  $a_j = 1$ , i.e., the client generates a new packet at the beginning of every uplink subslot with probability one. Now, we extend the above analysis to activity factors  $a_j < 1$ . Let  $\Pr\{b_n, l, m, n|b_o\}$  denote the probability that a given period with  $b_o$  backlogged packets in the beginning

- 1) has  $m$  bad slots and  $n$  good slots;
- 2) has  $l$  packet generations;
- 3) ends with a backlog of  $b_n$ ,  $b_n = 0, \dots, B_{\max,j}$  packets.

Toward the calculation of  $\Pr\{b_n, l, m, n|b_o\}$ , note that client  $j$  generates  $l$  new packets in a period of duration  $(m + n)$  slots with probability  $\binom{m+n}{l} a_j^l (1 - a_j)^{m+n-l}$ . Let  $g(n)$  denote the maximum number of successfully transmitted packets in  $n$  good slots. We have (12), shown at the bottom of the page. [The  $g(n)$  for other forms of SMPT is derived in analogous fashion.] Thus,

$$g(n) = \begin{cases} n \cdot R, & \text{for basic SMPT} \\ \min\left\{ \frac{n \cdot (n+1)}{2}, \frac{R \cdot (R+1)}{2} + R \cdot \max(n - R, 0) \right\}, & \text{for slow-healing SMPT.} \end{cases} \quad (12)$$

we have (13), shown at the bottom of the page. To see this, note that there are  $b_o + l$  packets to be transmitted in the period and that up to  $g(n)$  packets can be transmitted in the period. Also, note that at most one new packet is generated per slot; thus, we cannot have a situation in which a large burst of new packets arrive toward the end of a period and that this burst could not be cleared.

Finally, we obtain the transition probabilities  $\Pr\{b_n|b_o\}$  of the Markov Chain as

$$\Pr\{b_n|b_o\} = \sum_{m>0} \sum_{n>0} \sum_{l=0}^{m+n} \Pr\{b_n, l, m, n|b_o\}. \quad (14)$$

### B. Refined Analysis for Finite Buffer $B_{\max,j}$

We now refine our calculation of the buffer-occupancy distribution to account for a finite link-layer buffer capacity of  $B_{\max,j}$  packets. In contrast to the infinite buffer scenario analyzed above, with a finite buffer, arriving packets are lost when they find the buffer full. This, in turn, results in a smaller number of packets that are actually serviced. To account for this effect, we first find the number of backlogged packets in the finite buffer of capacity  $B_{\max,j}$  at the end of the run of bad slots in a period. Let  $k$  denote the number of packets generated during the run of bad slots and let  $b_{o,n}$  denote the number of backlogged packets at the end of the  $m$  consecutive bad slots in a given period. Recalling that  $b_o$  denotes the number of backlogged packets at the beginning of the run of bad slots (i.e., the beginning of the considered period), we clearly have

$$b_{o,n} = \max(b_o + k, B_{\max,j}).$$

If  $b_{o,n} = B_{\max,j}$ , then a packet that is generated (with probability  $a_j$ ) at the beginning of the first good slot of the run of  $n$  consecutive good slots is lost. To simplify the notation, we conservatively assume here that a packet is generated (with probability one) at the beginning of this first good slot. In Appendix B, we conduct an exact analysis with a packet generation with probability  $0 < a_j \leq 1$  in the first good slot. Let  $l$  denote the number of packets that are generated in the  $n - 1$  good slots (following the first good slot). Recalling that  $b_n$  denotes the backlog at the end of the run of good slots, (i.e., the end of the period), we have

$$b_n = \max\{\max(b_{o,n} + 1, B_{\max,j}) + l - g(n), 0\}.$$

To see this, note that  $\max(b_{o,n} + 1, B_{\max,j})$  packets are backlogged at the beginning of the run of good slots right after

the assumed packet generation at the beginning of the first good slot. Also, recall that  $g(n)$  denotes the maximum number of packets that are successfully transmitted in  $n$  good slots [see (12)]. We define  $\Pr\{b_n, k, l, m, n|b_o\}$  as the probability that given a backlog of  $b_o$  at the beginning of a period consisting of  $m$  bad slots followed by  $n$  good slots, with  $k$  packets generated in the  $m$  bad slots and  $l$  packets generated in the (last)  $n - 1$  good slots, we have  $b_n$  packets in the buffer at the end of the period. With the above definitions, it is clear that [see (15) at the bottom of the page].

From this, we obtain

$$\Pr\{b_n|b_o\} = \sum_{\forall n} \sum_{\forall m} \sum_{k=0}^m \sum_{l=0}^{n-1} \Pr\{b_n, k, l, m, n\}.$$

### C. Refined Analysis for Packet Drop in Good State

So far, we have assumed that the packet transmissions in good states are always successful, i.e., that  $q_{\text{good}}^j = 0$ . In order to incorporate a nonzero packet-drop probability  $q_{\text{good}}^j$  into our analysis, we approximate  $g(n)$  [the maximum number of successfully transmitted packets in  $n$  good slots, as given by (12)] by  $(1 - q_{\text{good}}^j) \cdot g(n)$  in (13) and the analysis that follows. As our numerical results in Section IV-E and [52] demonstrate, this approximation is highly accurate.

### D. Buffer Content at End of the Run of Bad Slots

Let  $P(B_j^b = b_{j,b})$ ,  $b_{j,b} = 0, \dots, B_{\max,j}$  denote the steady-state probability that client buffer  $j$  holds  $b_{j,b}$  packets at the end of the last bad slot of the run of consecutive bad slots of a period. We obtain  $P(B_j^b = b_{j,b})$  from  $P(B_j^g = b_j)$ , as follows. The conditional probabilities  $P(B_j^b = b_{j,b}|B_j^g = b_j)$  for  $0 \leq b_{j,b} < B_{\max}$  are calculated as shown in (16) at the bottom of the page, where  $\pi_b(m) = \sum_{\forall n>0} \pi(m, n)$ . From this, we calculate  $\Pr(B_j^b = b_{j,b})$  as

$$\begin{aligned} \Pr(B_j^b = b_{j,b}) &= \sum_{\forall m>0} \sum_{b_j=0}^{B_{\max}} \Pr(B_j^b = b_{j,b}, m|B_j^g = b_j) \cdot \Pr(B_j^g = b_j). \quad (17) \end{aligned}$$

After calculating these probabilities,  $\Pr(B_j^b = B_{\max})$  is given by  $1 - \sum_{b_{j,b}=0}^{B_{\max}-1} \Pr(B_j^b = b_{j,b})$ .

---


$$\Pr\{b_n, l, m, n|b_o\} = \begin{cases} \binom{m+n}{l} a_j^l (1 - a_j)^{m+n-l} \cdot \pi(m, n), & \text{if } b_n = \max(0, b_o + l - g(n)) \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$


---

$$\Pr\{b_n, k, l, m, n|b_o\} = \begin{cases} \binom{m}{k} a_j^k (1 - a_j)^{m-k} \cdot \binom{n-1}{l} a_j^l (1 - a_j)^{n-1-l} \cdot \pi(m, n), & \text{if } b_n = \max(0, \max(b_o + k + 1, B_{\max}) + l - g(n)) \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$


---

$$\Pr(B_j^b = b_{j,b}, m|B_j^g = b_j) = \begin{cases} \binom{m}{b_{j,b}-b_j} \cdot a_j^{b_{j,b}-b_j} (1 - a_j)^{m-b_{j,b}+b_j} \cdot \pi_b(m), & \text{if } b_{j,b} \geq b_j \text{ and } m \geq b_{j,b} - b_j \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

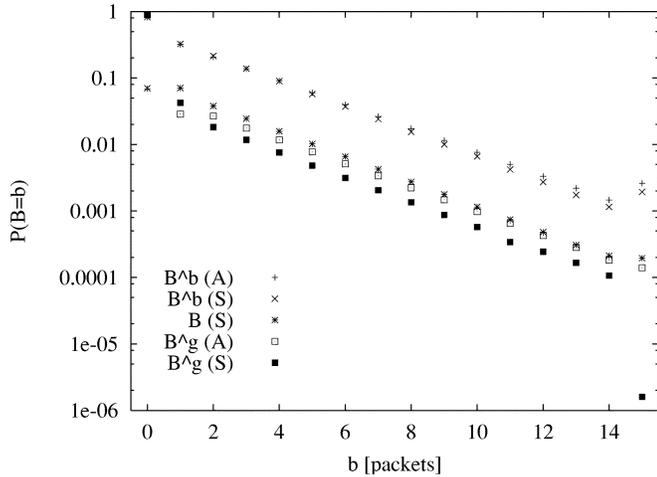


Fig. 9. Buffer occupancy probabilities for  $R_j = 2$ , activity factor  $a_j = 0.8$ , and probability of packet drop in good state  $q_{\text{good}}^j = 0.02$  and buffer size  $B_{\text{max},j} = 15$ .

### E. Numerical Results

We have conducted extensive numerical investigations and comparisons with simulations to verify the accuracy of our analytical results. In this section, we give a brief overview of these investigations and refer to [52] for more details. All numerical results presented in this paper are for slow-healing SMPT. In Fig. 9, we plot the probability masses  $P(B = b)$ ,  $P(B_j^g = b)$ , and  $P(B_j^b = b)$  for  $b = 0, 1, \dots, B_{\text{max},j}$ . The  $P(B_j^g = b)$  and  $P(B_j^b = b)$  are obtained both from our analysis (marked A) and simulation (marked S). The  $P(B = b)$  are obtained from simulation. All simulations are run until the 90% confidence intervals are less than 10% of the corresponding sample means. In the scenario considered, we set the packet-generation probability to  $a_j = 0.8$  and the packet-drop probability in the good state to  $q_{\text{good}}^j = 0.02$ . The channel-state transition probabilities are set to  $\alpha_j = 1/27$  and  $\beta_j = 1/3$  and the spreading gain is set to  $G = 64$ . The considered client has a buffer capacity of  $B_{\text{max},j} = 15$  packets and uses at most  $R_j = 2$  codes in parallel. We observe that the buffer occupancy probability masses generally drop off roughly linearly. The probability masses for  $P(B_j^b = b)$  and  $P(B_j^g = b)$  obtained from simulation have, in the mid range of buffer occupancies, an almost constant offset from  $P(B_j = b)$ . At the extreme ends of the buffer ( $b = 0$  and  $b = B_{\text{max},j} = 15$ ), they diverge more from  $P(B_j = b)$ , as discussed in detail in [52]. We observe that the analytical results for  $P(B_j^b = b)$  almost coincide with the corresponding simulation results for the entire range of  $b$ , with a very slight overestimation for  $b = B_{\text{max},j} = 15$ . We also observe that the analytical results for  $P(B_j^g = b)$  generally overestimate the corresponding simulation results. This is due to the approximation made in Section IV-C for the packet drop in the good slots. Interestingly, we observe that the analytical result for  $P(B_j^g = b)$  gives a fairly good approximation of the  $P(B_j = b)$  obtained from simulations, with some slight underestimation. Our more extensive numerical investigations in [52] indicate that these observations hold for a wide range of the system parameters.

In the following, we focus on the performance metrics-loss probability and throughput, as defined in Section II-A. In par-

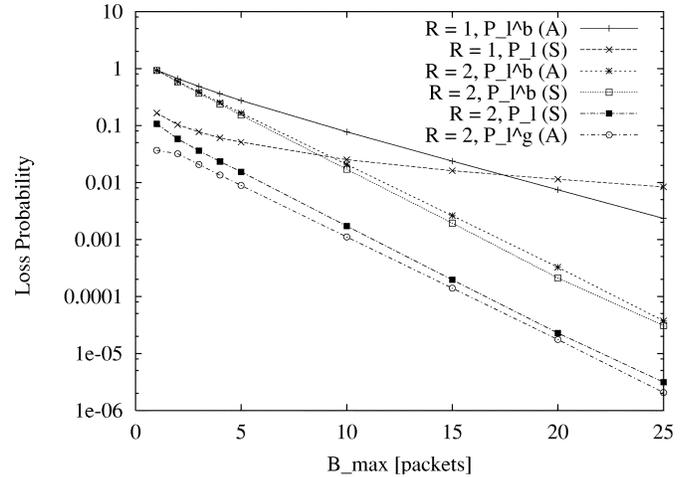


Fig. 10. Loss probability as a function of link-layer buffer capacity  $B_{\text{max},j}$  for conventional ARQ ( $R_j = 1$ ) and SMPT with  $R_j = 2$  ( $a_j = 0.8$ ,  $q_{\text{good}}^j = 0.02$ , fixed).

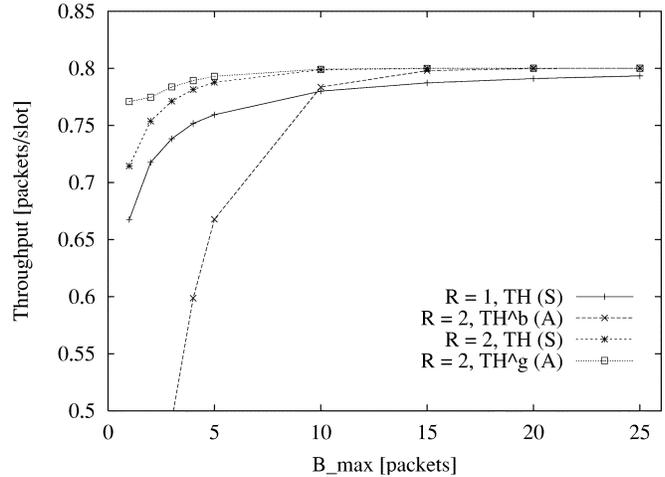


Fig. 11. Throughput of a client as a function of buffer capacity  $B_{\text{max},j}$  for conventional ARQ ( $R_j = 1$ ) and SMPT with  $R_j = 2$  ( $a_j = 0.8$ ,  $q_{\text{good}}^j = 0.02$ , fixed).

ticular, we focus on the actual loss probability  $P_l(j) = P(B_j = B_{\text{max},j})$  as obtained from simulation and the analytical estimates  $P_l^b = P(B_j^b = B_{\text{max},j})$  and  $P_l^g = P(B_j^g = B_{\text{max},j})$ ; for notational convenience, we define  $P_l := P_l(j)$ ,  $P_l^b := P(B_j^b = B_{\text{max},j})$ , and  $P_l^g := P(B_j^g = B_{\text{max},j})$  for the remainder of this section. Note that  $P_l$ ,  $P_l^g$ , and  $P_l^b$  are given by the right-most points (for  $b = B_{\text{max},j} = 15$ ) of the series of points marked “B (S),” “B<sup>b</sup> (A),” and “B<sup>g</sup> (A),” respectively, in Fig. 9. Similarly, we consider the actual (simulation) throughput  $\text{TH} = a_j \cdot [1 - P_l]$  and the analytical estimates  $\text{TH}^b = a_j \cdot [1 - P_l^b]$  and  $\text{TH}^g = a_j \cdot [1 - P_l^g]$ .

In Fig. 10, we plot the loss probability  $P_l$  (obtained from simulation) and the analytical estimates  $P_l^b$  and  $P_l^g$  as a function of link-layer buffer capacity  $B_{\text{max},j}$  in packets. Similarly, in Fig. 11, we plot the throughput  $\text{TH}$  (obtained from simulation) and the analytical estimates  $\text{TH}^b$  and  $\text{TH}^g$  (all in packet per slot) as a function of  $B_{\text{max},j}$  in packets. For both figures, the activity factor is  $a_j = 0.8$  and the packet-drop probability in the good state is  $q_{\text{good}}(j) = 0.02$ . We provide results for a

maximum number of  $R_j = 1$  and  $R_j = 2$  usable CDMA code channels. We observe from Fig. 10 that the SMPT mechanism with  $R_j = 2$  achieves significantly smaller loss probabilities than conventional ARQ with  $R_j = 1$ . For a buffer capacity of  $B_{\max,j} = 20$  packets, the loss probability  $P_l$  with  $R_j = 2$  is over two orders of magnitude smaller. We also observe that (for  $R_j = 2$ ) for the entire range of considered buffer capacities, the analytically obtained  $P_l^b$  overestimates the actual  $P_l$  by roughly one order of magnitude. The analytically obtained  $P_l^g$  underestimates the actual  $P_l$  very slightly. Given that loss probabilities are typically measured in orders of magnitude, our analytical results give reasonably good estimates of the actual loss probability for  $R_j = 2$ . The same holds for  $R_j = 3$  (see [52]).

For  $R_j = 1$ , our analytical results are not quite as good. We observe from Fig. 10 that our analytical estimate  $P_l^b$  does not always bound  $P_l$  from above. However, our analytical estimate is always within one order of magnitude of the actual loss probability  $P_l$ . We note that the focus of our analysis is on the SMPT mechanisms with  $R_j \geq 2$ , for which our analytical estimates are quite accurate. The conventional ARQ mechanisms with  $R_j = 1$  have been analyzed in the literature reviewed in Section I-A. Therefore, we focus on the analytical results for  $R_j \geq 2$  in the remainder of this paper and present only simulation results for  $R_j = 1$ .

We observe from Fig. 11 that for SMPT with  $R_j = 2$ , the client achieves throughputs close to the maximum of  $a_j = 0.8$  for a relatively small link-layer buffers of  $B_{\max,j} = 5$  packets. With conventional ARQ with  $R_j = 1$ , a significantly larger buffer is required to get close to the maximum throughput of 0.8. We note that all the results discussed here are for the link layer at a given client. The observed link-layer performance characteristics have important implications for the performance at the higher protocol layers, as discussed in Section VI-A

## V. ANALYSIS OF CHANNEL USAGE

In this section, we analyze the channel usage in the uplink subslots of a wireless CDMA system running some form of SMPT. We consider the independent link model throughout this section and the results of the analysis in this section are used in Section VI to analyze the clients' buffer contents in the interference link model.

### A. Channel Usage of an Individual Client

We first analyze the channel usage of an individual wireless client. Let  $C_j(a_j)$  be a discrete random variable denoting the number of CDMA codes used by client  $j$  in a given uplink subslot in steady state. In other words,  $C_j(a_j)$  denotes the number of simultaneous (parallel) packet transmissions from client  $j$  to the base station in a given (arbitrary) uplink subslot. Our goal is to calculate the distribution function

$$P(C_j(a_j) = c), \quad c = 0, \dots, R_j.$$

Initially, we analyzed the channel usage for an activity factor of  $a_j = 1$  and subsequently extended the analysis to activity factors  $a_j < 1$ .

For the analysis, we define a *c-code slot* for a given client as a slot in which the client uses  $c$  codes in the uplink subslot of the slot, i.e., transmits  $c$  packets in parallel to the base station. Suppose that a period consists of  $m$  bad slots followed by  $n$  good slots and suppose that client  $j$  has a backlog of  $b$  packets at the beginning of the period. Let  $\mathcal{N}_c(m, n, b)$  denote the number of  $c$ -code slots in that period for the client. In steady state, the probability that client  $j$  uses  $c$  codes in an uplink subslot is the ratio of the average number of  $c$ -code slots in a period to the average number of slots in a period. Formally

$$\begin{aligned} & P(C_j(1) = c) \\ &= \frac{\sum_{b=0}^{B_{\max,j}} \sum_{m,n>0} \mathcal{N}_c(m, n, b) \cdot \pi(m, n) \cdot \Pr(B_j^g = b)}{\bar{T}_j}, \\ & \quad c = 0, 1, \dots, R_j. \end{aligned} \quad (18)$$

We now proceed to calculate  $\mathcal{N}_c(m, n, b)$  for basic and slow-healing SMPT, described in Section III for  $c = 2, \dots, R_j$ ; other forms are analyzed analogously. First, we consider basic SMPT. With basic SMPT and  $a_j = 1$ , the number of codes used in an uplink subslot is equal to the number of backlogged packets (i.e., packets in the client buffer at the end of the preceding downlink subslot) plus one, since one new packet is generated at the beginning of the considered uplink subslot. However, at most  $R_j$  codes are used. The evaluation of  $\mathcal{N}_c(m, n, b)$  for basic SMPT is summarized in Table II. The main idea behind this analysis is to consider the different scenarios in which the number of backlogged packets and, thus, the code usage evolves over a period. Recall that  $\mathcal{N}_c(m, n, b)$  is defined as the number of times that  $c$  codes are used in a period with  $m$  bad slots,  $n$  good slots, and an initial backlog of  $b$  packets. For  $2 \leq c < R_j$ , there are up to two possibilities to use  $c$  codes: when “ramping up” and  $c$  codes are either used in the first bad slot (subcase I.1) or one of the subsequent bad slots (subcase I.2), and when  $c-1$  backlogged packets are left when clearing the backlog (subcase I.3). For  $c = R_j$ , we need to distinguish the scenarios in which  $R_j$  codes are used toward the end of the run of bad slots (i.e., after ramping up to  $R_j - 1$  backlogged packets during the first  $R_j - 1 - b$  bad slots, subcases II.2 and II.4) or during all of the bad slots (i.e., when there are  $R_j - 1$  or more backlogged packets at the beginning of the period, subcases II.1 and II.3). Additionally, we need to consider scenarios in which  $R_j - 1$  backlogged packets are cleared in each of the good slots (subcases II.1 and II.2) or where fewer than  $R - 1$  backlogged packets are transmitted in some of the good slots (subcases II.3 and II.4). With the definitions of the auxiliary variables  $\eta(m, b)$ , which is the number of times that the client needs to transmit with  $R_j$  codes in order to clear the backlog that has accumulated by the end of the run of consecutive bad slots, and  $\theta(m, b)$ , which is the residual backlog that requires fewer than  $R_j - 1$  codes to clear, and by using the standard impulse and unit-step functions,  $\mathcal{N}_c(m, n, b)$  can be summarized as stated in Table II.

For slow-healing SMPT, the analysis follows the same principles as the analysis for basic SMPT. However, the slow-healing SMPT analysis is somewhat more involved, as there is a larger number of scenarios to consider. We refer the reader to Appendix A for details.

TABLE II  
EVALUATION OF  $\mathcal{N}_c(m, n, b)$  FOR BASIC SMPT

Case I:	$2 \leq c < R_j$
Subcase I.1:	$b = c - 1$
Subcase I.2:	$b \leq c - 2$ and $m + b \geq R_j$
Subcase I.3:	$\text{mod}(m + b, R - 1) = c - 1$ and $n \geq \lfloor \frac{m+b}{R-1} \rfloor + 1$
$\mathcal{N}_c(m, n, b) =$	$\begin{cases} 2 & \text{if Subcases I.1 and I.3 hold,} \\ 2 & \text{if Subcases I.2 and I.3 hold,} \\ 1 & \text{if one of the Subcases I.1, I.2, or I.3 holds,} \\ 0 & \text{otherwise.} \end{cases}$
Case II:	$c = R_j$
Subcase II.1:	$b \geq R_j - 1$ and $\lfloor \frac{m+b}{R-1} \rfloor \geq n \Rightarrow \mathcal{N}_c(m, n, b) = m + n$
Subcase II.2:	$b < R_j - 1$ and $\lfloor \frac{m+b}{R-1} \rfloor \geq n \Rightarrow \mathcal{N}_c(m, n, b) = m - R_j + b + 1 + n$
Subcase II.3:	$b \geq R_j - 1$ and $\lfloor \frac{m+b}{R-1} \rfloor < n \Rightarrow \mathcal{N}_c(m, n, b) = m + \lfloor \frac{m+b}{R-1} \rfloor$
Subcase II.4:	$b < R_j - 1$ and $\lfloor \frac{m+b}{R-1} \rfloor < n \Rightarrow \mathcal{N}_c(m, n, b) = m - R_j + b + 1 + \lfloor \frac{m+b}{R-1} \rfloor$
Defn.:	$\theta(m, b) = \text{mod}(m + b, R_j - 1)$
	$\eta(m, b) = \lfloor \frac{m+b}{R_j-1} \rfloor$
	$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise.} \end{cases}$
	$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$
Summary:	
$\mathcal{N}_c(m, n, b) =$	$\begin{cases} \delta(b - R_j - 1) + u(c - 2 - b) \cdot u(m + b - c) + \delta(\theta(m, b) - c + 1) \cdot u(n - \eta(m, b) - 1) \\ \quad \text{if } 2 \leq c < R_j \\ \max(m - \max(R_j - 1 - b, 0), 0) + \min(\eta(m, b), n) \\ \quad \text{otherwise.} \end{cases}$

With the derived  $\mathcal{N}_c(m, n, b)$ ,  $c = 2, \dots, R_j$ , we evaluate  $P(C_j(1) = c)$ ,  $c = 2, \dots, R_j$  using (18) and finally

$$P(C_j(1) = 1) = 1 - \sum_{c=2}^{R_j} P(C_j(1) = c).$$

Note that the above analysis is for a client with an activity factor of  $a_j = 1$ . For a client with an activity factor  $0 < a_j < 1$ , we approximate the channel-usage distribution by  $P(C_j(a_j) = 0) = 1 - a_j$  and  $P(C_j(a_j) = c) = a_j \cdot P(C_j(1) = c)$ ,  $c = 1, 2, \dots, R_j$ .

### B. Channel Usage (Interference Level) in a Wireless Cell

We now analyze the total number of channels that are interfering with the transmissions of a given client in the considered wireless cluster/cell. We continue to consider the independent link model and the interference link model in the next section. Recall that there are  $J$  wireless clients in the cell and that the distribution of the number of used codes by each of the clients  $j$ ,  $j = 1, \dots, J$  is given by  $P(C_j(a_j) = c)$ ,  $c = 0, \dots, R_j$ , provided in the previous section.

Let  $j^*$  denote the considered client and note that the transmissions of the other clients  $j = 1, \dots, J$ ,  $j \neq j^*$  are interfering with the transmissions of client  $j^*$ . Let  $C$  be a discrete random variable denoting the total number of used CDMA codes by the interfering clients  $j = 1, \dots, J$ ,  $j \neq j^*$  in a given uplink subslot in steady state. Note that  $C$  gives the interference level for client  $j^*$  (in the number of used codes) in the cell. With

the independent link model, the distribution  $P(C = c)$ ,  $c = 0, \dots, \sum_{j \neq j^*} R_j =: R_{\text{tot}}$  is given by the convolution of the individual distributions  $P(C(a_j) = c)$ ,  $j = 1, \dots, J$ ,  $j \neq j^*$ . For notational convenience, we define  $\bar{C}$  as the average total number of interfering codes, i.e.,  $\bar{C} = \sum_{c=0}^{R_{\text{tot}}} c \cdot P(C = c)$ . Also, let  $\sigma_C^2$  denote the variance of the random variable  $C$ , i.e.,  $\sigma_C^2 = \sum_{c=0}^{R_{\text{tot}}} (c - \bar{C})^2 \cdot P(C = c)$ .

## VI. ANALYSIS OF BUFFER OCCUPANCY AT A WIRELESS CLIENT WITH INTERFERENCE LINK MODEL

In this section, we analyze the buffer occupancy of a wireless client  $j^*$  with the interference link model. This analysis takes the interference in the cell due to the uplink transmissions of all interfering clients  $j = 1, \dots, J$ ,  $j \neq j^*$  into consideration. The main idea of our analysis is to approximately capture the interference effect by basing the packet-drop probability in the good state on the mean and variance of the interference level. Specifically, we evaluate the bit-error probability as  $q_{\text{bit}}(\bar{C} + \tau \cdot \sigma_C)$  using (8), where  $\tau$ ,  $\tau > 0$  denotes the tuning parameter. We then evaluate the packet-drop probability in the good state of the interference link model as  $q_{\text{good}}(\bar{C} + \tau \cdot \sigma_C)$  using (11). Note that we approximate the number of interfering channels in a CDMA system running SMPT by  $\bar{C} + \tau \cdot \sigma_C$ . In other words, we use the sum of the means and standard deviations of the numbers of codes used by the individual clients in an interference-free setting to approximate the interference level in a realistic setting with interference. Our numerical work (see Section VI-A)

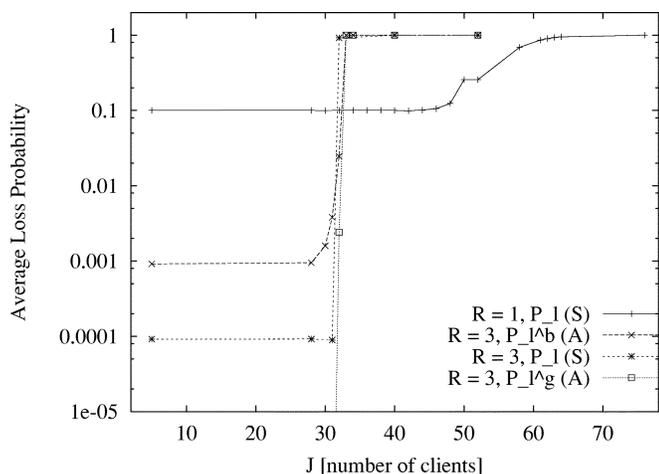


Fig. 12. Average loss probability as a function of a number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 3$  (activity factor  $a = 1$ ,  $B_{\max} = 20$  packets, fixed).

indicates that this approximation has good accuracy for  $\tau = 1$ . The packet-drop probability  $q_{\text{good}}(\bar{C} + \tau \cdot \sigma_C)$  is taken into account in the calculation of the buffer occupancy, as detailed in Section IV-C

#### A. Numerical Results

In this section, we present a representative sample of our extensive numerical investigations of the buffer occupancy in wireless clients in a multicode CDMA system running SMPT. In the scenarios presented here, all clients have a link-layer buffer capacity of  $B_{\max,j} = 20$  packets and the spreading gain is set to  $G = 64$ . We focus on slow-healing SMPT throughout this section. We focus on homogeneous clients in this section, i.e., all clients have the same activity factor ( $a_j = a \forall j = 1, \dots, J$ ) and maximum number of usable codes ( $R_j = R, \forall j = 1, \dots, J$ ). Our performance metrics are the actual average loss probability  $P_l$ , as defined in (1) and obtained from simulation, as well as the analytical estimates  $P_l^b := (1/J) \sum_{j=1}^J P(B_j^b = B_{\max,j})$  and  $P_l^g := (1/J) \sum_{j=1}^J P(B_j^g = B_{\max,j})$ . We also consider the corresponding aggregate throughput TH as defined in (3) and obtained from simulation, as well as the analytical estimates  $\text{TH}^b = \sum_{j=1}^J a_j \cdot (1 - P(B_j^b = B_{\max,j}))$  and  $\text{TH}^g = \sum_{j=1}^J a_j \cdot (1 - P(B_j^g = B_{\max,j}))$ . We also consider the average link-layer buffer occupancy in the clients  $B_{\text{avg}} = (1/J) \sum_{j=1}^J \sum_{b=1}^{B_{\max,j}} b \cdot P(B_j = b)$  and the standard deviation of the link-layer buffer occupancy  $B_{\text{std}}$ . In Figs. 12–21, we plot the average loss probability, aggregate throughput, average buffer occupancy, and standard deviation of the buffer occupancy as a function of the number of clients  $J$  in the cell for different combinations of the activity factor  $a$  and the number of usable codes  $R$ .

A number of observations are in order. Focusing for now on the plots of the average loss probability, we observe that, in its stable region, SMPT with  $R = 2$  or  $R = 3$  achieves a dramatically smaller average loss probability than conventional ARQ with  $R = 1$ . This difference is most pronounced for an activity factor of  $a = 1$ . As explained shortly, with conventional ARQ,

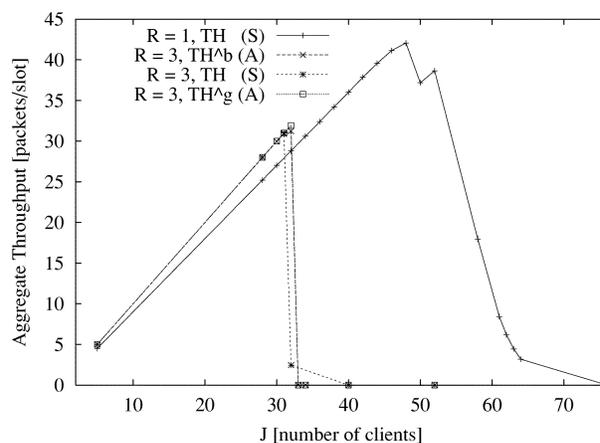


Fig. 13. Aggregate throughput in a cell as a function of number of supported clients  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 3$  (activity factor  $a = 1$ ,  $B_{\max} = 20$  packets, fixed).

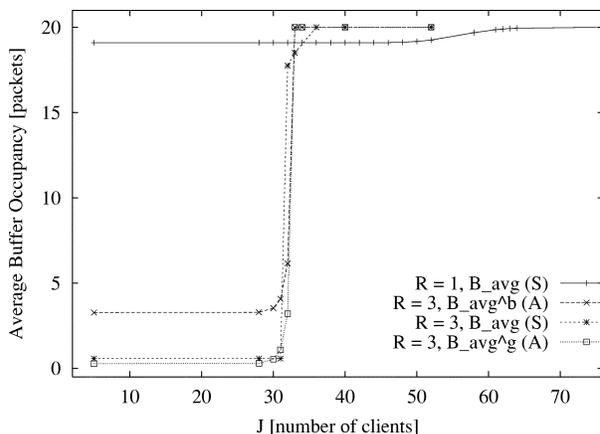


Fig. 14. Average buffer occupancy  $B_{\text{avg}}$  as a function of number of supported clients  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 3$  (activity factor  $a = 1.0$ ,  $B_{\max} = 20$  packets, fixed).

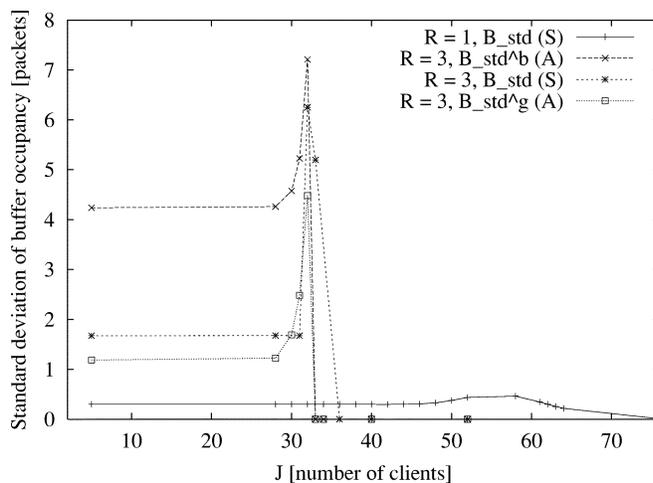


Fig. 15. Standard deviation of buffer occupancy  $B_{\text{std}}$  as a function of number of supported clients  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 3$  (activity factor  $a = 1.0$ ,  $B_{\max} = 20$  packets, fixed).

essentially all packets arriving during a client's bad channel state are lost. (In addition, as the number of clients  $J$  increases, the increasing interference level causes some packet drops in the good channel state, which in turn results in the loss of some

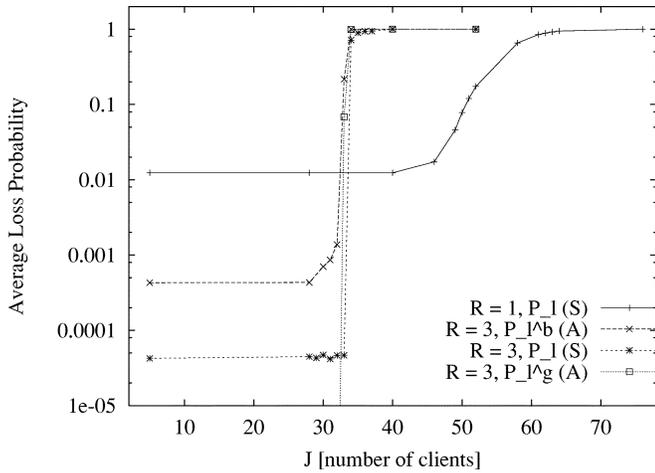


Fig. 16. Average loss probability as a function of number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 3$  (activity factor  $a = 0.9$ ,  $B_{\max} = 20$  packets, fixed).

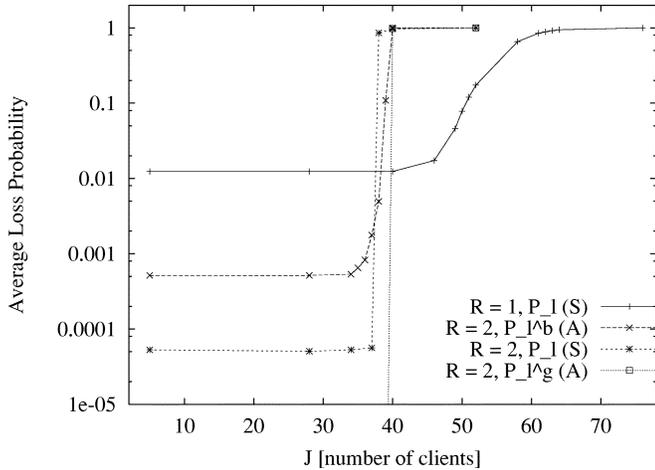


Fig. 17. Average loss probability as a function of number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (activity factor  $a = 0.9$ ,  $B_{\max} = 20$  packets, fixed).

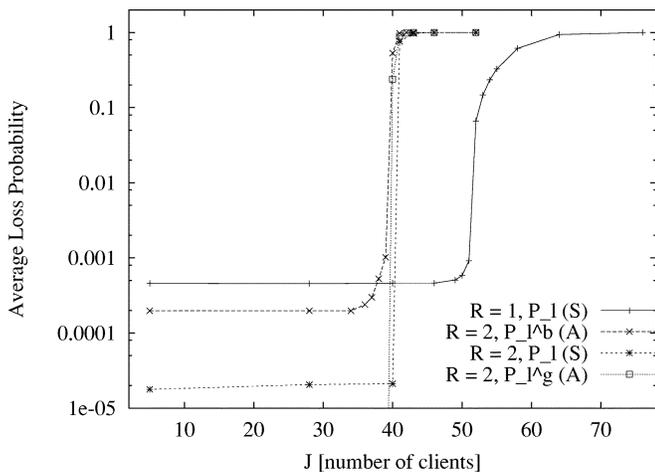


Fig. 18. Average loss probability as a function of number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (activity factor  $a = 0.8$ ,  $B_{\max} = 20$  packets, fixed).

arriving packets.) With a client's channel being in the bad state with a probability of  $P_{\text{bad}}^j = \alpha/(\alpha+\beta) = 0.1$  (in the considered scenario with  $\alpha = 1/27$  and  $\beta = 1/3$ ) and a new packet arriving

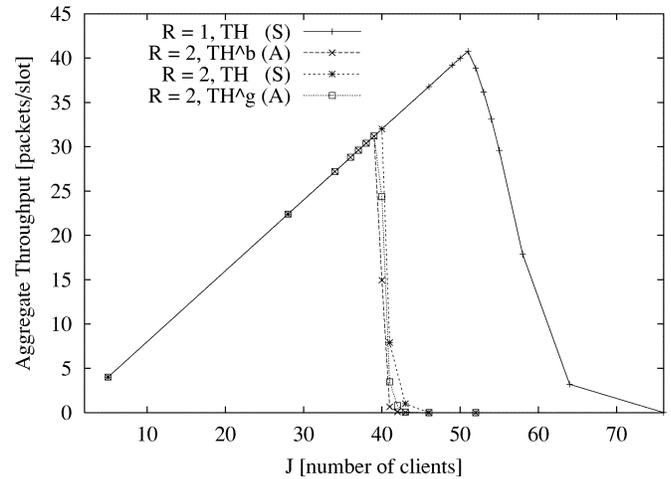


Fig. 19. Aggregate throughput in cell as a function of number of supported clients  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (activity factor  $a = 0.8$ ,  $B_{\max} = 20$  packets, fixed).

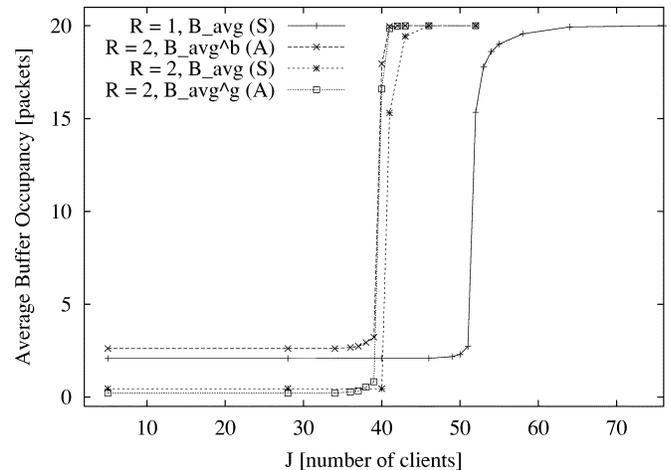


Fig. 20. Average buffer occupancy  $B_{\text{avg}}$  as a function of number of supported clients  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (activity factor  $a = 0.8$ ,  $B_{\max} = 20$  packets, fixed).

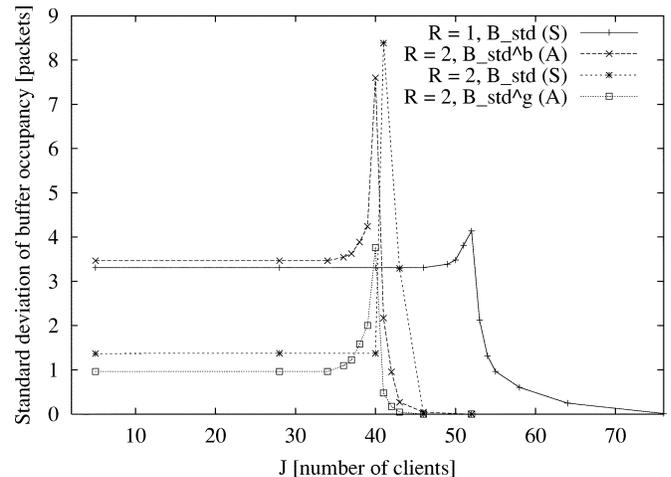


Fig. 21. Standard deviation of buffer occupancy  $B_{\text{std}}$  as a function of number of supported clients  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (activity factor  $a = 0.8$ ,  $B_{\max} = 20$  packets, fixed).

in every slot (with activity factor  $a = 1$ ), this results in an average loss probability of roughly 0.1 for low interference levels (small  $J$ ). With SMPT, the average loss probability for  $a = 1$

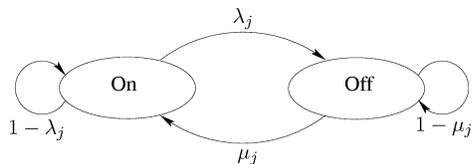


Fig. 22. Bursty traffic at the link layer of each client is modeled with an independent two-state Markov Chain.

and small  $J$  is roughly  $10^{-4}$ , i.e., approximately three orders of magnitude smaller than with conventional ARQ. As the activity factor decreases, this difference becomes less pronounced; for  $a = 0.8$ , the loss probability with SMPT is roughly one and a half orders of magnitude smaller than with conventional ARQ. This is primarily because, for small  $a$ , the (long-run) average utilization of the single uplink code channel used by conventional ARQ is smaller. Thus, conventional ARQ is more successful in absorbing the wireless channel variations. Note that this holds only for the nonbursty traffic considered here. For bursty traffic, there is a significant performance difference between SMPT and conventional ARQ, even for small long-run average channel utilization (see Section VII).

The second striking observation is that SMPT has very pronounced regions of stable and unstable operation. The average loss probability with SMPT typically jumps abruptly from values close to the loss probabilities obtained with the interference-free independent link model in Section IV to a value close to one. (When comparing the results in Fig. 10 with the results in Fig. 18, note that Fig. 10 gives results for a fixed  $q_{\text{good}} = 0.02$ , whereas for Fig. 18,  $q_{\text{good}}$  is a function of the interference level, i.e., the number of clients  $J$ , as described in Section II-B. For small  $J$ ,  $q_{\text{good}}$  is essentially zero in the interference link model, as can be seen from Fig. 4.) For large activity factors of  $a = 1.0$  and  $0.9$ , conventional ARQ ( $R = 1$ ) has a gradual transition from stable to unstable operation, i.e., exhibits a graceful degradation of link-layer QoS that may be interpreted as “soft capacity.” For smaller activity factors, e.g.,  $a = 0.8$ , conventional ARQ exhibits also a rather abrupt transition from stable to unstable operation. For a given activity factor  $a$  and maximum number of usable codes  $R$ , we refer to the largest number of clients that are supported in a stable fashion (i.e., the right-most point of the stable region in the loss-probability plots) as *capacity*. The abrupt jump of the loss probability of SMPT when its capacity is exceeded is explained as follows. SMPT strives to stabilize the throughput over the wireless link by adding additional CDMA codes in response to packet drops on the wireless link. This strategy works well as long as the system operates within its capacity. Notice from Fig. 13 how the aggregate SMPT throughput increases linearly (with a slope equal to the activity factor) up to the SMPT capacity (of 31 flows, in this case), whereas the throughput with conventional ARQ increases at a smaller slope. (Notice from Fig. 19 that for the  $a = 0.8$  scenario, the throughputs for both SMPT and conventional ARQ increase at the same slope, which is due to the smaller loss probabilities in this scenario.) Once the system capacity is exceeded, however, SMPT’s strategy to add more codes in response to packet drops becomes counter productive. The additional codes increase the interference level, which in turn increases the probability of

packet drop on the wireless links. The additional packet drops call for the use of even more codes and drive the system into instability. For this reason, SMPT systems need to be carefully dimensioned and managed (at the call/flow time scale; note that no coordination at all is required at the packet time scale). As we observe from the plots and discuss in more detail shortly, our analytical estimates track the system capacity with good accuracy and, thus, provide a basis for the dimensioning and management of SMPT systems.

We observe from the plots that SMPT trades off increased link-layer QoS for a reduced capacity. For instance, we see from Fig. 17 that SMPT with  $R = 2$  supports up to 37 clients with  $a = 0.9$  with an average loss probability of roughly  $5 \cdot 10^{-5}$ , whereas conventional ARQ with  $R = 1$  supports up to 46 such clients with an average loss probability of roughly  $2 \cdot 10^{-2}$ . As this example illustrates, at the expense of a moderate reduction in capacity, SMPT provides significantly reduced link-layer buffer occupancy and buffer overflow (loss). Two points are important when interpreting this result. First, although we do not explicitly study the packet delay and packet jitter, our results for the buffer occupancy give a rough indication of the packet delays. We observe from Fig. 14 that for  $a = 1$  (and small  $J$ ) the link-layer buffer holds on average  $B_{\text{avg}} = 19$  packets with conventional ARQ ( $R = 1$ ). With a UMTS packet slot length of 10 ms, this translates into a delay of (at least) 190 ms. SMPT, on the other hand, has on average (roughly) one packet in the buffer, which results in much smaller delays. The small standard deviation of the buffer occupancy for conventional ARQ with  $a = 1$  in Fig. 15 is due to the buffer typically holding either 19 or 20 packets and never being drained to lower levels. (This is also why essentially all packets arriving in bad slots are lost.) SMPT has less than two packets of standard deviation, indicating that it provides reasonably small levels of packet-delay variation (jitter) in addition to the small average delay depicted in Fig. 14. We observe from Fig. 20 that for a smaller activity factor of  $a = 0.8$ , conventional ARQ also achieves smaller average buffer occupancies of  $B_{\text{avg}} = 2$  packets, which are, however, still larger than with SMPT (which tends to keep the buffer completely empty with an average occupancy of  $B_{\text{avg}} = 0.2$  packets in this scenario). We observe from Fig. 21 that in this lower activity factor scenario, SMPT achieves a significantly smaller standard deviation of the buffer occupancy, of  $B_{\text{std}} = 1.3$  packets versus 3.3 packets with conventional ARQ, which indicates a smaller delay jitter for SMPT.

The second aspect to keep in mind is that the link-layer performance analyzed in this paper has important implications on the performance of the higher protocol layers and the application. Although multimedia applications are expected to mostly run over user datagram protocol (UDP), some of these applications are likely to run over transport control protocol (TCP), e.g., see [3], [53], and [54]. The large link-layer losses, as well as the large delays and delay jitters of conventional ARQ, tend to trigger frequent TCP retransmissions and cuts into TCP’s congestion window, resulting in an overall degradation of TCP throughput. SMPT, on the other hand, achieves small losses, delays, and delay jitters at the link layer, i.e., SMPT “hides” to a large extent the unreliability of the wireless link from TCP, which is a good strategy for improving TCP throughput [55],

[56]. With UDP, the transport layer is less affected by link-layer delays and losses. However, the performance of multimedia applications, including voice, video, and interactive gaming, is typically severely degraded by large link-layer delays and losses, as well as large delay variations.

Returning to the performance of SMPT at the link layer, we observe by comparing Figs. 16 and 17 that SMPT with  $R = 3$  gives very slightly smaller loss probabilities and a slightly smaller capacity than SMPT with  $R = 2$ . This indicates that a low-cost mobile terminal with support for up to two simultaneous transmissions can extract most of the gain in link-layer QoS from SMPT.

In additional experiments, we have compared basic, fast-healing, and slow-healing SMPT. We have observed that basic SMPT gives marginally smaller loss probabilities than the other schemes considered at the expense of a smaller (by typically around 10%) capacity. Fast-healing SMPT, in turn, gives a very slightly smaller loss probability and capacity than does slow-healing SMPT.

We observe from the figures that  $P_l^b$  gives a conservative analytical estimate of the actual loss probability in the stable region. Importantly,  $P_l^b$  tracks the jump to the unstable operation very precisely and, thus, provides an accurate analytical characterization of the capacity. We also observe that for small  $J$ , the analytical estimate  $P_l^g$  gives very small loss probabilities, which do not show up on the plots. This is because, in the good state of the interference link model and for small  $J$ , it is exceedingly rare that subsequent packets are dropped on the wireless link. The estimate  $P_l^g$  jumps to one very close to the actual capacity (obtained from simulation), with a slight over estimation for  $a = 0.9$  and  $R = 2$ . Overall, our numerical results indicate that the analytical estimate  $P_l^b$  allows for an accurate analytical characterization of the flow carrying capacity of a multicode CDMA system running some form of SMPT. The estimate  $P_l^b$  may, thus, provide a basis for dimensioning and resource management in SMPT systems, as well as the further optimization of SMPT.

## VII. EFFECTS OF BURSTY TRAFFIC

The numerical results of the previous section demonstrate that SMPT can significantly improve the link-layer QoS for nonbursty traffic, especially when the activity factor  $a$  is close to one, i.e., when the long-run average utilization of the single code channel used by conventional ARQ is large. In this section, we examine the regime where the long-run average-channel utilization is small. Of particular interest in this regime is bursty traffic. Bursty traffic at the link layer arises in practice due to the combination of bursty application traffic [57], [58] and the fragmentation of higher layer protocol data units (IP datagrams) into smaller link-layer protocol data units (referred to as packets in this paper).

Following [9] and [10], we model the bursty traffic at the link layer of each client  $j$ ,  $j = 1, \dots, J$  by an independent two-state Markov chain with the states "ON" and "OFF," as illustrated in Fig. 22. These Markov chains make state transitions at the end of every downlink subslot. The state-transition probabilities are denoted by  $\lambda_j$  and  $\mu_j$ , as illustrated in Fig. 22. A client in the ON state generates a new packet at the beginning of an uplink subslot with probability one, whereas in the OFF state, no new packets are generated. Note that the long-run average probability of packet

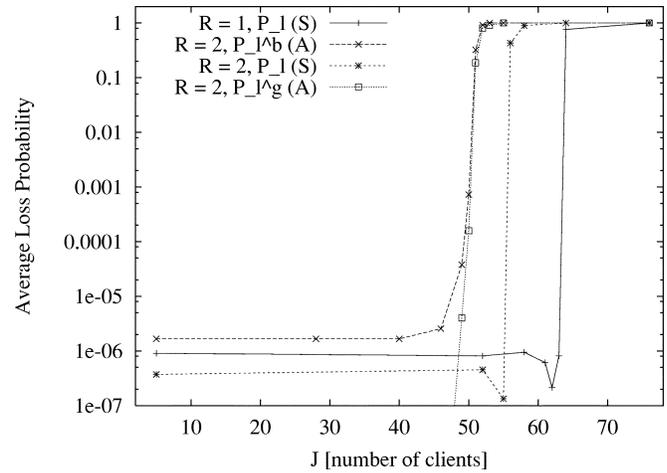


Fig. 23. Average loss probability as a function of number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (nonbursty traffic with activity factor  $a = 0.5$ , the arrivals being Bernoulli process,  $B_{\max} = 20$  packets, fixed).

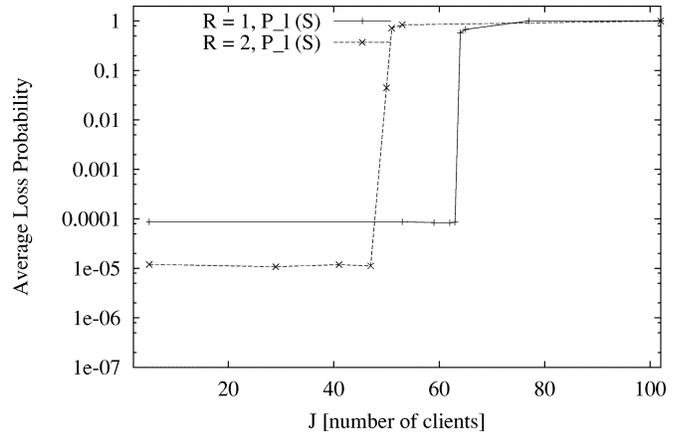


Fig. 24. Average loss probability as a function of number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (bursty traffic with  $\mu/(\lambda + \mu) = 0.5$  and  $1/\lambda = 10$ ,  $B_{\max} = 20$  packets, fixed).

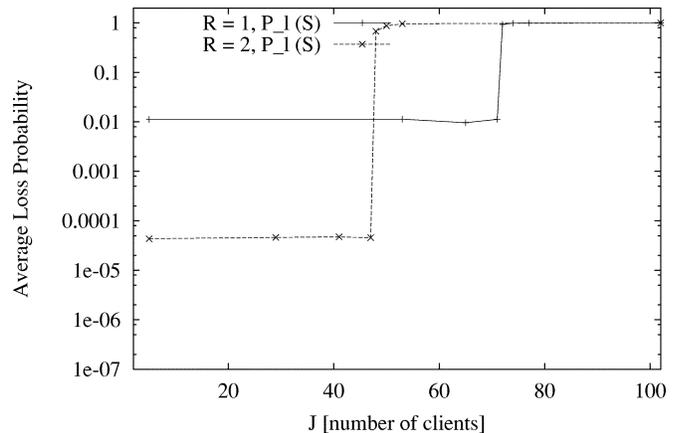


Fig. 25. Average loss probability as a function of number of supported clients (flows)  $J$  for conventional ARQ ( $R = 1$ ) and SMPT with  $R = 2$  (bursty traffic with  $\mu/(\lambda + \mu) = 0.5$  and  $1/\lambda = 100$ ,  $B_{\max} = 20$  packets, fixed).

generation in a slot is  $\mu_j/(\lambda_j + \mu_j)$ . Also, note that the average burst length of  $1/\lambda_j$  packets increases as  $\lambda_j$  decreases. In other words, for small  $\lambda_j$ , the traffic becomes more bursty.

In Figs. 23–25, we give a sample of our comparisons of the average loss probabilities achieved by SMPT and conventional

ARQ for nonbursty and bursty traffic, we refer the interested reader to [52] for more details. In all cases depicted here, the long-run average packet-generation probability in a slot is set to 0.5, i.e.,  $a_j = 0.5$  for the nonbursty traffic and  $\mu_j/(\lambda_j + \mu_j) = 0.5$  for the bursty traffic for all clients  $j$ ,  $j = 1, \dots, J$ . We observe from Fig. 23 that with this low utilization of its single code channel, conventional ARQ achieves loss probabilities almost as small as SMPT for nonbursty traffic. For bursty traffic, on the other hand, SMPT achieves significantly smaller loss probabilities, as Figs. 24 and 25 indicate. The gap in performance widens as the traffic becomes more bursty; for an average burst size of 100 packets, the loss probability of SMPT is over two orders of magnitude smaller. This improved link-layer QoS of SMPT comes again at the expense of a smaller capacity (number of supported flows). Overall, our results indicate that within its smaller capacity range, SMPT effectively stabilizes the link-layer QoS by absorbing the variations of the wireless channels, as well as the variations of the bursty traffic.

### VIII. CONCLUSION

We have studied SMPT, which is a novel class of ARQ mechanisms for multicode CDMA systems. We found that SMPT significantly stabilizes the throughput and reduces delay and jitter over wireless links, which are key issues in enabling high-quality multimedia services over wireless links. The improved link-layer QoS of SMPT comes at the expense of a moderate reduction of the number of flows that are supported in a shared interference environment (cell, cluster). We have developed an analytical framework that accurately characterizes the tradeoff between the improved link-layer QoS and the reduced number of flows that are provided with this higher QoS. Thus, our analytical framework provides a basis for the resource management in wireless systems running SMPT.

Our analytical framework is modular in structure, giving insights into the key effects that govern the behavior and performance of SMPT. The modular structure of our analytical framework allows for the characterization of new forms of SMPT, i.e., new policies for dynamically adding CDMA code channels in response to packet drops. Thus, our framework may serve as a platform for developing new forms of SMPT and optimizing these for given wireless communication settings.

The analytical framework developed in this paper is limited to nonbursty Bernoulli traffic. In ongoing work, we are extending the framework to more general bursty traffic patterns. The presented analytical framework is furthermore limited to single-rate traffic scenarios in which each client generates traffic that can be carried by a single CDMA code (provided that there are no link errors). In ongoing work, we are extending the framework to multirate traffic scenarios in which some high-speed clients require multiple CDMA codes (even if there are no link errors). We expect the SMPT analysis modules developed in this paper to be useful building blocks for the analysis of more general traffic scenarios. There are several other avenues for future work on SMPT, such as extensions of the analytical framework to more general packet service disciplines at the link-layer and delay metrics.

Another exciting research direction is to study cross-layer designs involving SMPT at the link layer. We envision that SMPT may be combined with adaptive FEC to form hybrid SMPT techniques analogous to the extensively studied hybrid ARQ techniques. In addition, the physical layer infrastructure may be exploited for more efficient link probing. We also envision that advanced forms of SMPT may exploit information from the higher protocol layers and application (e.g., play out deadlines of video frames) for more efficient packet scheduling.

### APPENDIX A

#### EVALUATION OF $\mathcal{N}_c(m, n, b)$ FOR SLOW-HEALING SMPT

*Case I:*  $2 \leq c < R_j$  and  $m + b \leq R_j(R_j - 1)/2$

Subcase I.1:  $b = c - 1$

Subcase I.2:  $n \geq c$  and  $\eta_r(m, b) \geq c$

Subcase I.3:  $b + m - \{\eta_r(m, b) (\eta_r(m, b) - 1) / 2\} = c - 1$   
and  $n \geq \eta_r(m, b) + 1$

$\mathcal{N}_c(m, n, b)$

$$= \begin{cases} 3, & \text{if subcases I.1, I.2, and I.3 hold,} \\ 2, & \text{if two of the subcases I.1, I.2, and I.3 hold,} \\ 1, & \text{if one of the subcases I.2, I.2, and I.3 holds.} \end{cases}$$

*Case II:*  $c = R_j$  and  $m + b \leq R_j(R_j - 1)/2$

Subcase II.1:  $b \geq R_j - 1$  and  $m + b < R \cdot (R - 1)/2 \Rightarrow \mathcal{N}_c(m, n, b) = 1.$

Subcase II.2:  $b \geq R_j - 1$ ,  $m + b = R \cdot (R - 1)/2$  and  $n \geq R \Rightarrow \mathcal{N}_c(m, n, b) = 2.$

Subcase II.3:  $b < R_j - 1$ ,  $m + b = R \cdot (R - 1)/2$  and  $n \geq R \Rightarrow \mathcal{N}_c(m, n, b) = 1.$

Subcase II.4:  $b < R_j - 1$  and  $m + b < R \cdot (R - 1)/2 \Rightarrow \mathcal{N}_c(m, n, b) = 0.$

Subcase II.5:  $b < R_j - 1$ ,  $m + b = R \cdot (R - 1)/2$  and  $n < R \Rightarrow \mathcal{N}_c(m, n, b) = 0.$

*Case III:*  $2 \leq c < R_j$  and  $m + b > R_j(R_j - 1)/2$

Subcase III.1:  $b = c - 1.$

Subcase III.2:  $n \geq c.$

Subcase III.3:  $\theta_s(m, b)$  and  $n \geq \eta_s(m, b).$

$\mathcal{N}_c(m, n, b)$

$$= \begin{cases} 3, & \text{if subcases III.1, III.2, and III.3 hold,} \\ 2, & \text{if two of the subcases III.1, III.2, and III.3 hold,} \\ 1, & \text{if one of the subcases III.2, III.2, and III.3 holds.} \end{cases}$$

*Case IV:*  $c = R_j$  and  $m + b > R_j(R_j - 1)/2$

Subcase IV.1:  $b \geq R_j - 1$  and  $n \geq R + \eta_s(m, b) \Rightarrow \mathcal{N}_c(m, n, b) = \eta_s(m, b) + 2.$

Subcase IV.2:  $b \geq R_j - 1$  and  $R \leq n < R + \eta_s(m, b) \Rightarrow \mathcal{N}_c(m, n, b) = n - R + 2.$

Subcase IV.3:  $b \geq R_j - 1$  and  $n < R \Rightarrow \mathcal{N}_c(m, n, b) = 1.$

Subcase IV.4:  $b < R_j - 1$  and  $n \geq R + \eta_s(m, b) \Rightarrow \mathcal{N}_c(m, n, b) = \eta_s(m, b) + 1.$

Subcase IV.5:  $b < R_j - 1$  and  $R \leq n < R + \eta_s(m, b) \Rightarrow \mathcal{N}_c(m, n, b) = n - R + 1.$

Subcase IV.6:  $b < R_j - 1$  and  $n < R \Rightarrow \mathcal{N}_c(m, n, b) = 0.$

$$\mathcal{N}_c(m, n, b) = \begin{cases} \delta(b - c + 1) + u(n - c) \cdot u(\eta_r(m, b) - c) + \delta(\theta_r(m, b) - c + 1), & \text{in Case I} \\ u(b - R_j + 1) + \delta(\eta_r(m, b) - R_j) \cdot u(n - R_j), & \text{in Case II} \\ \delta(b - c + 1) + u(n - c) + \delta(\theta_s(m, b) - c + 1) \cdot u(n - R - \eta_s(m, b) - 1), & \text{in Case III} \\ u(b - R + 1) + \min(1 + \eta_s(m, b), \max(n - R + 1, 0)), & \text{in Case IV} \end{cases}$$

$$\Pr\{b_n, k, l, m, n, a_s | b_o\} = \begin{cases} \binom{m}{k} a_j^k (1 - a_j)^{m-k} \cdot a_j \cdot \binom{n-1}{l} a_j^l (1 - a_j)^{n-1-l} \cdot \pi(m, n), & \text{if } b_n = \max\{0, \max(b_o + k + 1, B_{\max}) + l - 1 - g(n)\} \text{ and } a_s = 1 \\ \binom{m}{k} a_j^k (1 - a_j)^{m-k} \cdot (1 - a_j) \cdot \binom{n-1}{l} a_j^l (1 - a_j)^{n-1-l} \cdot \pi(m, n), & \text{if } b_n = \max\{0, \max(b_o + k, B_{\max}) + l - 1 - g(n)\} \text{ and } a_s = 0 \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

Definitions:

$$\eta_r(m, b) = \left\lfloor \frac{1 + \sqrt{1 + 8(b + m)}}{2} \right\rfloor$$

$$\theta_r(m, b) = b + m - \frac{\eta_r \cdot (\eta_r - 1)}{2}$$

$$\eta_s(m, b) = \left\lfloor \frac{b + m - \frac{R \cdot (R-1)}{2}}{R - 1} \right\rfloor$$

$$\theta_s(m, b) = \text{mod}\left(b + m - \frac{R \cdot (R-1)}{2}, R - 1\right)$$

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise.} \end{cases}$$

$$u(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Summary: See the equation at the top of the page.

## APPENDIX B

### EXACT ANALYSIS FOR FINITE BUFFER $B_{\max, j}$

In this appendix, we provide the exact analysis for the case of a finite buffer  $B_{\max, j}$ . The analysis in Section IV-B conservatively assumed that a new packet is generated with probability one in the first good slot of the run of  $n$  consecutive good slots. We now conduct an exact analysis with a packet generation with probability  $0 < a_j \leq 1$  in the first good slot. Let  $a_s$  be an indicator variable, which is 1 if a packet is generated in the first slot and 0 otherwise. The backlog at the end of the run of good slots is now given by

$$b_n = \begin{cases} \max\{\max(b_{o, n} + 1, B_{\max, j}) + l - g(n), 0\}, & \text{if } a_s = 1 \\ \max\{b_{o, n} + l - g(n), 0\}, & \text{otherwise.} \end{cases} \quad (19)$$

We extend the definition of  $\Pr\{b_n, k, l, m, n | b_o\}$  given in Section IV-B to  $\Pr\{b_n, k, l, m, n, a_s | b_o\}$ , which is given by (20), shown at the top of the page. Finally, we obtain

$$\Pr\{b_n | b_o\} = \sum_{\forall n} \sum_{\forall m} \sum_{k=0}^m \sum_{l=0}^{n-1} \sum_{a_s=0}^1 \Pr\{b_n, k, l, m, n, a_s | b_o\}.$$

## ACKNOWLEDGMENT

The authors are grateful to A. Wolisz for his support of this work and for hosting M. Krishnam at the Telecommunication Networks (TKN) Institute, Technical University Berlin, Berlin, Germany, in the summer of 2001.

## REFERENCES

- [1] "Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems," EIA/TIA-95 Rev. B, 1997.
- [2] "Universal Mobile Telecommunications System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS," UMTS 30.03.
- [3] F. Fitzek, R. Supratio, A. Wolisz, M. Krishnam, and M. Reisslein, "Capacity and QoS for streaming video applications over TCP in CDMA based networks," in *Proc. Int Conf. Third Generation Wireless and Beyond*, San Francisco, CA, May 2002, pp. 755–760.
- [4] F. Fitzek, M. Reisslein, and A. Wolisz, "Uncoordinated real-time video transmission in wireless multi-code CDMA systems: An SMPT-based approach," *IEEE Wireless Commun. Mag.*, vol. 9, pp. 100–110, Oct. 2002.
- [5] D. Ayyagari and A. Ephremides, "Cellular multicode CDMA capacity for integrated (voice and data) services," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 928–938, May 1999.
- [6] C. S. Kang and D. K. Sung, "Capacities of spectrally overlaid single-code and multicode CDMA systems," *IEEE Trans. Veh. Technol.*, vol. 51, pp. 839–854, Sept. 2002.
- [7] J. Y. Kim, G. L. Stuber, and I. F. Akyildiz, "A simple performance/capacity analysis of multiclass macrodiversity CDMA cellular systems," *IEEE Trans. Commun.*, vol. 50, pp. 304–308, Feb. 2002.
- [8] D. K. Kim and D. K. Sung, "Capacity estimation for a multicode CDMA system with SIR-based power control," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 701–710, May 2001.
- [9] J. G. Kim and M. M. Krunz, "Bandwidth allocation in wireless networks with guaranteed packet loss performance," *IEEE Trans. Networking*, vol. 8, pp. 337–349, June 2000.
- [10] —, "Delay analysis of selective repeat ARQ for a Markovian source over a wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1968–81, Sept. 2000.
- [11] M. M. Krunz and J. G. Kim, "Fluid analysis of delay and packet discard performance for QoS support in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 384–395, Feb. 2001.
- [12] N. Shacham and D. Towsley, "Resequencing delay and buffer occupancy under selective repeat ARQ with multiple receivers," *IEEE Trans. Commun.*, vol. 39, pp. 928–936, June 1991.
- [13] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 445–455, May 1997.

- [14] M. Elaoud and P. Ramanathan, "Adaptive use of error-correcting codes for real-time communication in wireless networks," *Proc. IEEE INFOCOM '98*, pp. 548–555, Apr. 1998.
- [15] H. Liu, H. Ma, M. El Zarki, and S. Gupta, "Error control schemes for networks: An overview," *Mobile Networks Applicat.*, vol. 2, no. 2, pp. 167–182, 1997.
- [16] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the gaussian collision channel," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1971–1988, July 2001.
- [17] P. Lettieri, C. Fragouli, and M. B. Srivastava, "Low power error control for wireless links," *Proc. ACM/IEEE MobiCom '97*, pp. 139–150, 1997.
- [18] H. Minn, M. Zeng, and V. K. Bhargava, "On ARQ scheme with adaptive error control," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 1426–1436, Nov. 2001.
- [19] S. Choi and K. G. Shin, "A class of adaptive hybrid ARQ schemes for wireless links," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 777–790, Mar. 2001.
- [20] S.-H. Hwang, B. Kim, and Y.-S. Kim, "A hybrid ARQ scheme with power ramping," in *Proc. IEEE Veh. Technol. Conf.*, 2001, pp. 1579–1583.
- [21] H. Liu and M. El Zarki, "Performance of H.263 video transmission over wireless channels using hybrid ARQ," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1775–1786, Dec. 1997.
- [22] I. Joe, "An adaptive hybrid ARQ scheme with concatenated FEC codes for wireless ATM," *Proc. ACM/IEEE MobiCom '97*, pp. 131–138, 1997.
- [23] K. Miyoshi, A. Matsumoto, C. Wengerter, M. Kasapidis, M. Uesugi, and O. Kato, "Constellation rearrangement and spreading code rearrangement for hybrid ARQ in MC-CDMA," in *Proc. IEEE Int. Symp. Wireless Pers. Multimedia Commun.*, 2002, pp. 668–672.
- [24] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless LAN's," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 524–534, June 2002.
- [25] S. Aramvith, C.-W. Lin, S. Roy, and M.-T. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 558–565, June 2002.
- [26] P.-C. Hu, Z.-L. Zhang, and M. Kaveh, "Channel condition ARQ rate control for real-time wireless video under buffer constraints," in *Proc. IEEE Int. Conf. Image Processing*, 2000, pp. 124–127.
- [27] M. Shiwen, L. Shunan, S. S. Panwar, and Y. Wang, "Reliable transmission of video over ad-hoc networks using automatic repeat request and multipath transport," in *Proc. IEEE Veh. Technol. Conf.*, 2001, pp. 615–619.
- [28] G. Wu, H. Harada, K. Taira, and Y. Hase, "An integrated transmission protocol for broadband mobile multimedia communication systems," in *Proc. IEEE Veh. Technol. Conf.*, 1997, pp. 1346–1350.
- [29] M.-S. Do, Y. Park, and J.-Y. Lee, "Channel assignment with QoS guarantees for a multiclass multicode CDMA system," *IEEE Trans. Veh. Technol.*, vol. 51, pp. 934–948, Sept. 2002.
- [30] M. R. Hueda, C. Rodriguez, and C. Marques, "Enhanced-performance video transmission in multicode CDMA wireless systems using a feedback error control scheme," *Proc. IEEE GLOBECOM*, pp. 619–626, Nov. 2001.
- [31] P.-R. Chang and C.-F. Lin, "Wireless ATM-based multicode CDMA transport architecture for MPEG-2 video transmission," *Proc. IEEE*, vol. 87, pp. 1807–1824, Oct. 1999.
- [32] J. He, M. T. Liu, Y. Yang, and M. E. Muller, "A MAC protocol supporting wireless video transmission over multi-code CDMA personal communication networks," *Comput. Commun.*, vol. 21, no. 14, pp. 1256–1268, 1998.
- [33] H. Liu, M. J. Karol, M. ElZarki, and K. Y. Eng, "Channel access and interference issues in multi-code DS-SS-CDMA wireless packet (ATM) networks," *Wireless Networks*, vol. 2, no. 3, pp. 173–192, 1996.
- [34] Z. Liu, M. Karol, M. ElZarki, and K. Eng, "Distributed-queueing request update multiple access (DQRUMA)," in *Proc. IEEE Int. Conf. Commun. '96*, Seattle, WA, June 1995, pp. 1224–1231.
- [35] J. Chin-Lin and S. Nanda, "Load and interference based demand assignment (LIDA) for integrated services in CDMA wireless systems," in *Proc. IEEE GLOBECOM*, Nov. 1996, pp. 235–241.
- [36] R. P. Ejzak, D. N. Knisely, S. Kumar, S. Laha, and S. Nanda, "BALI: A solution for high speed CDMA data," *Bell Labs. Tech. J.*, vol. 2, no. 3, pp. 134–151, 1997.
- [37] J. Sadowsky, Voice and Data Traffic Over the WCDMA Air Interface, presented at Arizona State Univ.
- [38] F. Wegner, personal communication, Mobile Radio Access Simulation Group, Siemens AG, Berlin, Germany, May 2000.
- [39] M. Zorzi, "Data-link packet dropping models for wireless local communications," *IEEE Trans. Veh. Technol.*, vol. 51, pp. 710–719, July 2002.
- [40] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1266, Sept. 1960.
- [41] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, Sept. 1963.
- [42] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LAN's," *ACM Wireless Networks*, vol. 3, pp. 91–102, 1997.
- [43] H. S. Wang and N. Moayeri, "Finite-state Markov model – A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 163–177, Feb. 1995.
- [44] M. Zorzi and R. R. Rao, "Error-constrained error control for wireless channels," *IEEE Pers. Commun.*, vol. 4, pp. 27–33, Dec. 1997.
- [45] M. Zorzi, R. R. Rao, and L. B. Milstein, "On the accuracy of a first-order Markovian model for data block transmission on fading channels," in *Proc. IEEE Int. Conf. Universal Pers. Commun.*, Nov. 1995, pp. 211–215.
- [46] —, "Error statistics in data transmissions over fading channels," *IEEE Trans. Commun.*, vol. 46, pp. 1468–77, Nov. 1998.
- [47] R. R. Rao, "Higher layer perspectives on modeling the wireless channel," in *Proc. IEEE Inform. Theory Workshop*, 1998, pp. 137–138.
- [48] G. L. Stüber, *Principles of Mobile Communications*, 2nd ed. Norwell, MA: Kluwer, 2001.
- [49] J. Holtzman, "A simple, accurate method to calculate spread spectrum multiple access error probabilities," *IEEE Trans. Commun.*, vol. 40, pp. 461–464, Mar. 1992.
- [50] L. Kleinrock, *Queueing Systems*. New York: Wiley, 1975, vol. 1.
- [51] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Norwell, MA: Kluwer, 1999.
- [52] M. Krishnam, M. Reisslein, and F. Fitzek. (2003) An analytical framework for simultaneous MAC packet transmission (SMPT) in a multi-code CDMA wireless system (extended version). Dept. Elect. Eng., Arizona State Univ, Tempe, AZ. [Online]. Available: <http://www.fulton.asu.edu/~mre>
- [53] C. Krasic, K. Li, and J. Walpole, "The case for streaming multimedia with TCP," in *Proc. 8th Int. Workshop Interactive Distributed Multimedia Systems*, Lancaster, U.K., Sept. 2001.
- [54] A. Goel, C. Krasic, K. Li, and J. Walpole, "Supporting low-latency TCP-based media streams," in *Proc. IEEE Int. Workshop Quality Service*, Miami Beach, FL, May 2002.
- [55] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Networking*, vol. 5, pp. 756–769, Dec. 1997.
- [56] H. M. Chaskar, T. V. Lakshman, and U. Madhoo, "TCP over wireless with link level error control: Analysis and design methodology," *IEEE/ACM Trans. Networking*, vol. 7, pp. 605–615, Oct. 1999.
- [57] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, Feb. 1994.
- [58] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Trans. Commun.*, vol. 43, pp. 1566–1579, Feb. 1995.



**Manjunath Krishnam** (S'99) received the B.E. degree in electronics and communications from R.V. College of Engineering, Bangalore University, Bangalore, India, in 1996 and the M.S. degree in electrical engineering from Arizona State University (ASU), Tempe, in 1999. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, ASU.

His research interests are in the areas of network performance analysis, network and traffic modeling, and resource management in wireless networks.



**Martin Reisslein** (A'96-S'97-M'98-SM'03) received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Dieburg, Germany, in 1994 and the M.S.E. and Ph.D. degrees from the University of Pennsylvania, Philadelphia, both in electrical engineering, in 1996 and 1998, respectively.

From July 1998 to October 2000, he was a Scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin, Germany. While in Berlin, he taught courses on performance evaluation and computer networking at

the Technical University Berlin. He currently is an Assistant Professor in the Department of Electrical Engineering, Arizona State University, Tempe. His research interests are in the areas of Internet quality of service, video-traffic characterization, wireless networking, and optical networking.

From 1994 to 1995, he visited the University of Pennsylvania as a Fulbright Scholar. He is the editor-in-chief of *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS* and has served on the technical program committees of *IEEE INFOCOM*, *IEEE GLOBECOM*, and the *IEEE International Symposium on Computer and Communications*. He was corecipient of the Best Paper Award of the *SPIE Photonics East 2000, Terabit Optical Networking* conference. He has organized sessions at the *IEEE Computer Communications Workshop (CCW)* and maintains an extensive library of video traces for network performance evaluation, including frame-size traces of MPEG-4 and H.263 encoded video, at <http://trace.eas.asu.edu>.



**Frank Fitzek** (M'98) received the diploma (Dipl.-Ing. degree) in electrical engineering from the University of Technology–Rheinisch–Westfälische Technische Hochschule (RWTH), Aachen, Germany, in 1997 and the Ph.D. (Dr.-Ing.) in electrical engineering from the Technical University Berlin, Berlin, Germany, in 2002.

He cofounded the start-up company *aticom GmbH* in Berlin in 1999, where he is currently involved in the development of advanced wireless IP services for WLAN and 3G. Since 2002, he has been

teaching and conducting research as a Postdoctoral Fellow at the University of Ferrara, Ferrara, Italy. Since 2003, he has been affiliated with the University of Aalborg, Aalborg, Denmark, as an Associate Professor. His current research interests are in the areas of QoS support for voice and video over wireless networks, robust header compression, security in self organizing networks, and the integration of mobile *ad hoc* networks in cellular systems.

Dr. Fitzek serves on the editorial board of *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*.