An Analytical Framework for Simultaneous MAC Packet Transmission (SMPT) in a Multi-Code CDMA Wireless System (Extended Version)

Manjunath Krishnam Martin Reisslein Frank Fitzek

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, which transmit one packet at a time over the wireless air interface, SMPT exploits the parallel code channels provided by multi-code CDMA. 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. We analyze the link layer buffer occupancy and the code usage in a wireless system running some form of SMPT. Our analysis quantifies the trade off 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.

Keywords

ARQ, Buffer Occupancy, Capacity, CDMA, Link Layer QoS, Multi-code, 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 wireless links. Typically, wireless systems employ a combination of automatic repeat request (ARQ) mechanisms and forward error correction

An overview of this work was presented at the *High-Speed Networking Workshop*, San Francisco, CA, March 2003.

Supported in part by the National Science Foundation through grant Career ANI-0133252 and the State of Arizona through the IT301 initiative.

Please direct correspondence to M. Reisslein. M. Krishnam and M. Reisslein are with the Dept. of Electrical Eng., Arizona State University, Goldwater Center, MC5706, Tempe, AZ 85287-5706, USA, phone:(480)965-8593, fax:(480)965-8325, e-mail: {manjunath,reisslein}@asu.edu, web: http://www.eas.asu.edu/~mre.

F. Fitzek is with acticom GmbH, Am Borsigturm 42, 13507 Berlin, Germany, e-mail: fitzek@acticom.de, web: http://www.acticom.de.

in conjunction with adaptive coding/modulation and power control to stabilize the wireless links. In this paper we focus on the ARQ component which 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 FEC and physical layer techniques. The ARQ mechanisms that are employed in wireless systems are typically based on one of the 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 re-transmitted 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 UMTS [2], are *multi-code* 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 multi-code 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 multi-code 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 multi-media 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, we develop in this paper an analytical frame work for SMPT, which 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., measurements of the signal to interference and noise ratio 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 terminalsi.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 build-up at the link layer, at the expense of increased interference. (We assume throughout that the used CDMA codes are correlated pseudo-noise 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 build-up (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 trade off between the increased link layer quality of service and the reduced capacity when running some form of SMPT in a multi-code CDMA system.

This paper is organized as follows. In the following subsection we review related work. In Section II we give an overview of the considered multi-code CDMA system. We 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, V, and 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], [6], [7], [8], [9], [10], [11], [12], [13], [14]. In contrast, in this paper, we study the performance of wireless systems at the radio link (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 [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. Also, the link layer performance of forward error correction (FEC) based schemes have been studied, see for instance [31], [32], [33].

Recently, hybrid ARQ schemes that combine some form of FEC with ARQ packet retransmissions have received considerable interest, see e.g., [34], [35], [36], [37], [38], [39], [40], [41]. Many of these hybrid ARQ schemes adapt to channel variations, e.g., by adjusting the FEC coding rate, packet length, etc., see for instance [42], [43], [44], [45], [46], [47], [48], [49]. 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/re-transmission) 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 LAN is developed in [50]. A hybrid ARQ scheme for video transmission over a single wireless channel is studied in [51]. An adaptive video encoder is embedded in the sender packet transmission-receive acknowledgment feedback loop in [52]; the encoder dynamically adjusts the encoding rate based on acknowledgment from the receiver. An interesting variation of ARQ is studied in [53] for transmitting two-layer encoded video over ad-hoc wireless networks. In this scheme, the base layer and the enhancement layer are transmitted over disjoint (multi-hop) paths. If a base layer is lost, then it is retransmitted over the enhancement layer path.

Closer related to our work are the studies on providing link layer QoS in multi-code CDMA sytstems. A conceptual framework for a multi-code CDMA based wireless multimedia network is described in [54]. This work points to the possibility of conducting ARQ retransmissions over a separate CDMA code channel, but does not analyze this approach. Admission control strategies for multiple traffic classes (each requiring a different, but fixed number of code channels) in a multi-code CDMA system are studied in [55]. A hybrid ARQ scheme for transmitting video in a multi-code CDMA system is developed and analyzed in [56]. 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 multi-code CDMA systems is also studied in [57]. The scheme proposed in [57] is similar to ours in that multiple codes are used in parallel on a dynamic basis. The main difference between [57] and SMPT is that [57] 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 multi-code CDMA systems with a fixed infra-structure and a *central* base-station, see for instance [58], [59] (which are based on the DQRUMA [60]) or the LIDA/BALI approach [61], [62]. 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 Multi-Code 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 Figure 1. Let J denote the number of wireless (and possibly mobile) terminals transmitting to the base-station. We consider a multi-code CDMA system where the base-station allocates to each wireless terminal, a set of unique pseudo noise code sequences for uplink transmission. Let $R(j),\ j = 1, \ldots, J$, denote the number of parallel code channels supported by the radio front-end of terminal j and suppose that the base-station

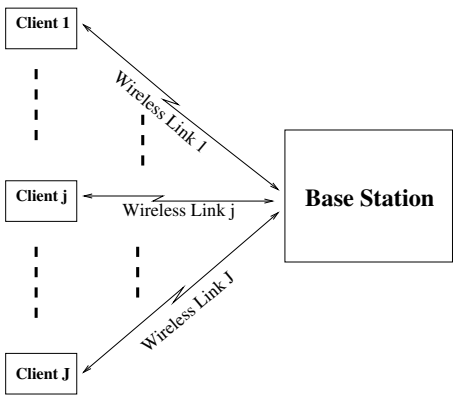

Fig. 1. System Architecture: J wireless clients conduct uncoordinated uplink transmissions to a base-station.

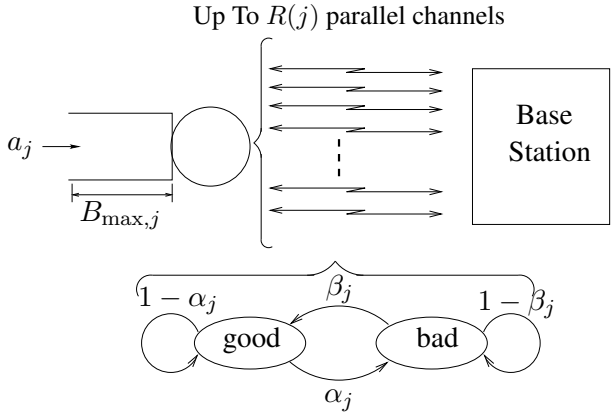

Fig. 2. The client buffer of capacity $B_{\max,j}$ packets is served by up to $R(j)$ parallel code channels. The wireless link (consisting of up to $R(j)$ channels) is modeled with a two-state Markov chain.

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 [63]). We consider a system with a time division duplex (TDD) timing structure. Specifically, time is divided into fixed-length *slots*. Each slot is sub-divided 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 non-bursty 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, we focus on a scenario where clients generate at most one packet per slot in this paper. Multi-rate traffic scenarios where 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 Figure 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 hardware 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 [64].) Each acknowledged packet is immediately flushed from the client's buffer. A packet which was transmitted in an uplink slot but is not acknowledged in the subsequent downlink slot, stays in the buffer. The client will attempt to re-transmit 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 pre-specified delay bound (see e.g., [65]).

Let $B_j,\ j = 1, \ldots, J$, 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 $P(B_j = b_j),\ b_j = 0, \ldots, B_{\max,j}$, 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

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

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

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

in packets per slot. We define the aggregate throughput 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.,

$$TH = \sum_{j=1}^{J} TH(j), \tag{3}$$

in packets per slot.

B. Wireless Link Model

We employ the widely used two-state Markov Chain model [66], [67] (also referred to as 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 on 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 [68], [69], [70], [71], [72], [30]. 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 [73]. 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 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.

B.1 Model Parameters

We denote α_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 β_j for link j's transition probability from the bad state to the good state. We denote P^j_{good} (P^j_{bad}) for the steady-state probability that link j is in the good (bad) state in a given slot, i.e., $P^j_{good} = \beta/(\alpha+\beta)$ and $P^j_{bad} = \alpha/(\alpha+\beta)$. We denote $\overline{T}^j_{good}$ ($\overline{T}^j_{bad}$) for link j's average sojourn time in the good (bad) state in slots, i.e., $\overline{T}^j_{good}$ ($\overline{T}^j_{bad}$) is the average number of consecutive good (bad) slots of link j. Clearly, $\overline{T}^j_{good} = 1/\alpha$ slots and $\overline{T}^j_{bad} = 1/\beta$ slots. We note that for a flat-fading channel, the Markov Chain parameters may be derived in terms of the Rayleigh fading parameters [74]. The steady-state probabilities are given in terms of the ratio of the Rayleigh fading envelope to the local root mean square level by $P^j_{good} = e^{-\rho_j^2}$ (and $P^j_{bad} = 1 - e^{-\rho_j^2}$). Typical values of the fade margin at the radio front-end of the wireless terminal are between 5 and 20 dB (i.e., ρ_j is typically between -20 and -5 dB). This corresponds to typical values between 0.9 and 0.999 for P^j_{good} and values between 0.001 and 0.1 for P^j_{bad}. We conservatively consider $P^j_{good} = 0.9$ and $P^j_{bad} = 0.1$ 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 λ 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 $\overline{T}^j_{bad} = 32$ msec. For our numerical work in this paper we consider a slot length of 10 msec and $\beta = 1/3$.

B.2 Period

As groundwork for our analytical framework, we analyze the lengths of the runs of consecutive good slots and consecutive 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 run of consecutive bad slots followed by a run of consecutive good slots. Let T^j_{good} (T^j_{bad}) be a discrete random variable denoting the number of consecutive good slots (bad slots) in a given period of link j. Clearly,

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

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

Note that by the Markovian property, T^j_{bad} and T^j_{good} 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 $\pi_j(m,n)$, is given by

$$\pi_j(m,n) \quad = \quad P(T^j_{bad} = m,\ T^j_{good} = n) \tag{6}$$

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

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

C. Packet Drop Probability in Good/Bad State

Throughout we set the packet drop probability in the bad state to $q^j_{bad} = 1$, 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^j_{good}. 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 $i,\ i = 0,\ldots,(J-1)\cdot R$, denote the total number of currently interfering pseudo-noise CDMA codes in the cell. (We assume here that the codes of a given client are orthogonal, achieved for instance by sub-code concatenation, such that there is no self-interference.) We employ the widely used Holtzman approximation [75] to calculate the bit error probability $q_{bit}(i)$ resulting from an interference level of i codes,

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

where (8)

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

and

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

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 $q_{bit}(i)$ we calculate the packet drop probability in the good state $q_{good}(i)$ by considering a simple static FEC as follows. We set the packet length to 1023 bits and employ static forward error correction that can correct up to 30 bit errors. (We use these settings to fix ideas, our analytical approach is valid for arbitrary settings of these parameters.) Thus,

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

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*. Figures 3 and 4 depict the bit error probability q_{bit} and the packet drop probability q_{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

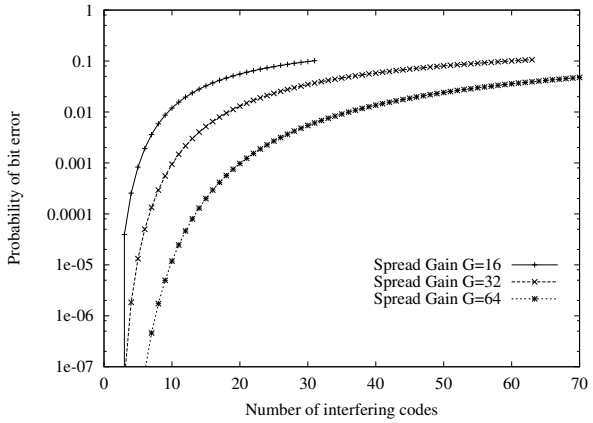

Fig. 3. Bit error probability $q_{bit}(i)$ as a function of total number of interfering codes i for different spreading gains G.

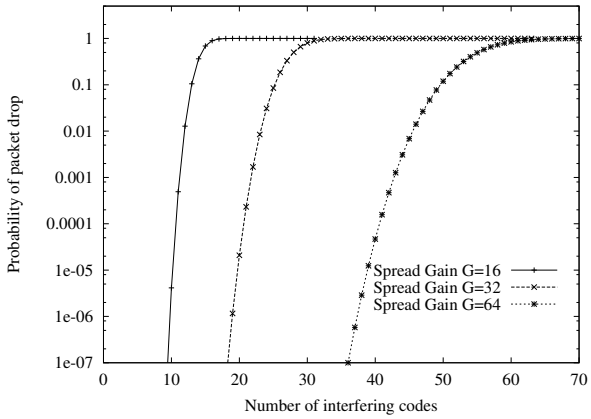

Fig. 4. Packet drop probability $q_{good}(i)$ as a function of total number of interfering codes i for different spreading gains G.

probability and the static FEC for the packet 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_{good} as a function of the number of interfering codes i.

III. Overview of Simultaneous MAC Packet Transmission (SMPT)

In this section we introduce Simultaneous MAC Packet Transmission (SMPT), a novel class of ARQ mechanisms for multi-code 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 re-transmits the packet until it is successfully transmitted (and acknowledged), as illustrated in Figure 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 build-up at the link layer in the client. This buffer build-up increases the probability of losing a newly generated packet due to buffer overflow, which we analyze in this paper. The buffer build-up 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.

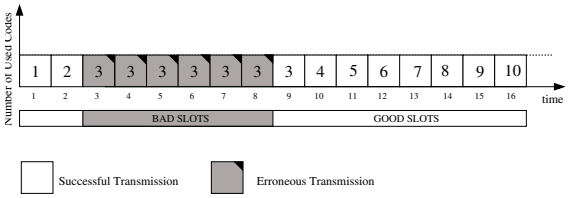

Fig. 5. Conventional ARQ: Packet drops on wireless link result in throughput fluctuations and buffer build-up at the link layer.

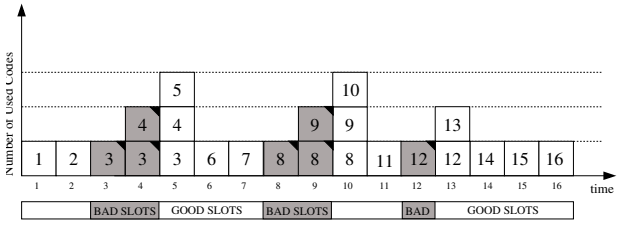

Fig. 6. Basic Simultaneous MAC Packet Transmission (SMPT): strives to stabilize throughput over wireless link and avoid buffer build-up.

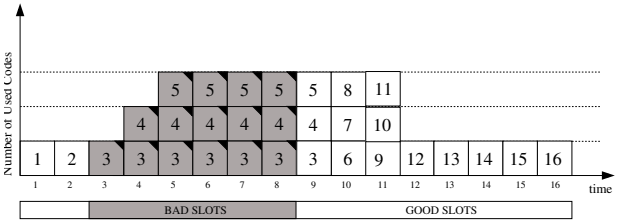

Fig. 7. Inefficiency of basic SMPT in a scenario where link errors are correlated.

The Simultaneous MAC Packet Transmission (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 Figure 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 pre-specified 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 Figure 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 incorporate *link probing* are introduced. If a transmitted packet is not acknowledged, the client re-transmits 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 Figure 8. With *fast-healing SMPT*, on the other hand, the client uses R CDMA codes in the slot right after the probing packet was successfully acknowledged, as illustrated in Figure 9. With both strategies, 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 (i) a bad slot or (ii) the independent packet drops with probability $q_{good}(i)$ 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.

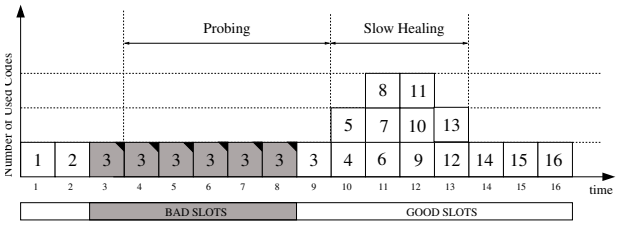

Fig. 8. Slow-Healing SMPT.

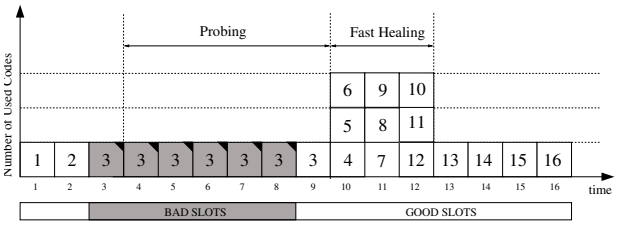

Fig. 9. Fast-Healing SMPT.

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 pre-specified probability after the successful probing packet is received. And then ramp up to three codes with a pre-specified 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 which can accommodate any possible form of SMPT, and thus provide a basis for further exploration and optimization.

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-station, as illustrated in Figure 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 $P(B_j = b_j),\ b_j = 0, \ldots, B_{\max,j}$, that client j holds b_j packets at the end of a down link subslot. In order not to 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., $B_{\max,j} = \infty$. We subsequently refine the analysis to finite buffers $B_{\max,j}$. In addition, we initially assume that packets are not dropped in the good link state, i.e., $q_{good} = 0$, 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 $P(B_j^g = b_j), b_j = 0, \ldots, B_{\max,j}$. 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 $P(B_j^g = b_j), b_j = 0, \ldots, B_{\max,j}$, is to construct an irreducible, positive recurrent, discrete-time Markov chain with the states $B_j^g = 0, \ldots, B_j^g = B_{\max,j}$. The Markov chain makes state transitions at the end of each period.

Let $Pr\{b_n|b_o\},\ b_n, b_o = 0, \ldots, B_{\max,j}$, 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 non-zero backlog of $b_n,\ b_n = 1, \ldots, B_{\max,j}$, packets at the end of the period given that the backlog at the beginning of the period is $b_o,\ b_o = 0, \ldots, B_{\max,j}$, packets. These probabilities are given in Table I for the different forms of 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 Eqn. (5).) The $Pr\{b_n, m|b_o\}$ for other forms would be derived in analogous fashion, and then incorporated in 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 $\max(b_o + m, B_{\max,j})$ backlogged packets at the end of the run of bad slots. This is because 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

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

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

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 where 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 fast-healing SMPT, one additional good slot (compared to basic SMPT) is required to clear a backlog of $b_o + m - b_n$ packets. This is because the first good slot of the run of good slots is used for probing; thus in this first good slot one packet is successfully transmitted and one packet is generated resulting in no effective reduction of backlog.

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 $b_o + m - b_n - R \cdot (R - 1)/2$ 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 $Pr\{b_n|b_o\},\ b_o, b_n = 0, \ldots, B_{\max,j}$, we find the steady-state probabilities $P(B_j^g = b_j),\ b_j = 0, 1, \ldots, B_{\max,j}$, using standard techniques [76], [77].

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 (i) has m bad slots and n good slots, (ii) has l packet generations, and (iii) ends with a backlog of $b_n,\ b_n = 0, \ldots, B_{\max,j}$, packets. Towards 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 m good slots. We have

$$g(n) = \begin{cases} n \cdot R & \text{for Basic SMPT} \\ (n-1) \cdot R & \text{for Fast Healing SMPT} \\ \min\{n \cdot (n+1)/2,\ R \cdot (R+1)/2 + R \cdot \max(n-R,\ 0)\} & \text{for Slow Healing SMPT.} \end{cases} \tag{12}$$

(The $g(n)$ for other forms of SMPT is derived in analogous fashion.) Thus,

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

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 where a large burst of new packets arrive towards the end of a period, and that burst could not be cleared.

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

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

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 C 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, clearly,

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

From this we obtain

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

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 $q^j_{good} = 0$. In order to incorporate a non-zero packet drop probability q^j_{good} into our analysis, we approximate $g(n)$ (the maximum number of successfully transmitted packets in n good slots, as given by (12)) by $(1 - q^j_{good}) \cdot g(n)$ [which we conservatively round down to the nearest integer] in (13) and the analysis that follows. As our numerical results in Section IV-E demonstrate, this approximation is highly accurate.

D. Buffer Content at End of Run of Bad Slots

Let $P(B^b_j = b_{j,b}),\ b_{j,b} = 0, \ldots, B_{\max,j}$, 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 $P(B^b_j = b_{j,b})$ from $P(B^g_j = b_j)$ as follows. The conditional probabilities $P(B^b_j = b_{j,b}|B^g_j = b_j)$ for $0 \leq b_{j,b} < B_{\max,j}$ are calculated as

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

where, $\pi_{b,j}(m) = \sum_{\forall n>0} \pi_j(m,n)$. From this we calculate $Pr(B^b_j = b_{j,b})$ as

$$Pr(B^b_j = b_{j,b}) = \sum_{\forall m>0} \sum_{b_j=0}^{B_{\max,j}} Pr(B^b_j = b_{j,b}, m|B^g_j = b_j) \cdot Pr(B^g_j = b_j). \tag{17}$$

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

E. Numerical Results

We have conducted extensive numerical investigations and comparisons with simulations to verify the accuracy of our analytical results. All numerical results presented in this paper are for slow-healing SMPT. In Figure 10 we plot the probability masses $P(B = b)$, $P(B^g_j = b)$ and $P(B^b_j = b)$ for $b = 0, 1, \ldots, B_{\max,j}$. The $P(B^g_j = b)$ and $P(B^b_j = b)$ 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 considered scenario we set the packet generation probability to $a_j = 0.8$ and the packet drop probability in the good state to $q^j_{good} = 0.02$. 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 $B_{\max,j} = 15$ 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 $P(B^b_j = b)$ and $P(B^g_j = b)$ 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

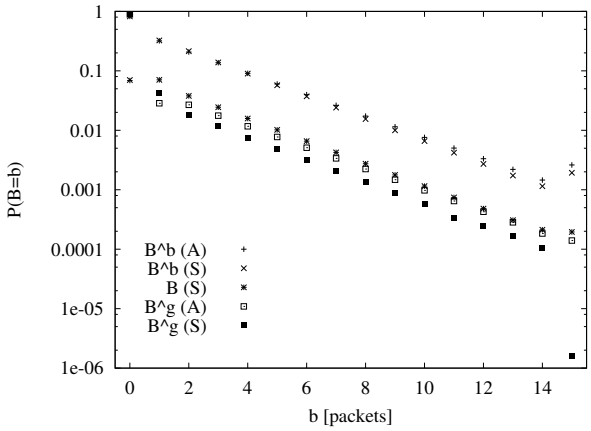

Fig. 10. Buffer occupancy probabilities for $R_j = 2$, activity factor $a_j = 0.8$, probability of packet drop in good state $q^j_{good} = 0.02$ and buffer size $B_{\max,j} = 15$.

$b = B_{\max,j} = 15$) they diverge more from $P(B_j = b)$. To explain the behavior of $P(B^b_j = b)$ for $b = 0$, note that the probability of the buffer being empty at the end of the run of bad slots is equal to the probability that there are no packets generated during these slots, which is given by

$$P(B^b_j = 0) = \sum_{\forall m>0} (1 - a_j)^m \cdot \pi_{b,j}(m). \tag{18}$$

To explain the behavior of $P(B^b_j = b)$ for $b = B_{\max,j}$, note that this probability mass "collects" the tail of the distribution $\pi_{b,j}(m)$. Specifically, suppose that there are b_{st} backlogged packets at the beginning of the run of consecutive bad slots, then

$$P(B^b_j = B_{\max,j}) = \sum_{m \geq B_{\max,j} - b_{st}} \pi_{b,j}(m) \tag{19}$$

To explain the simulation results for $P(B^g_j = b)$ for $b = B\max, j$, note that the buffer is full at the end of the run of good slots only if the buffer is full at the end of the preceding run of bad slots and every good slot in the run experiences packet drops, i.e.,

$$P(B^g_j = B_{\max,j}) = P(B^b_j = B_{\max,j}) \cdot \sum_{n>1} (q^j_{good})^{g(n)} \cdot \pi_{g,j}(n), \tag{20}$$

where $\pi_{g,j}(n) = \sum_{\forall m>0} \pi_j(m, n)$. The analytical result for $P(B^g_j = B_{\max,j})$ is almost two orders of magnitude larger than the actual $P(B^g_j = B_{\max,j})$ obtained from simulation. This is due to the conservative rounding down of $(1 - q^j_{good}) \cdot g(n)$ in Section IV-C.

We observe that the analytical results for $P(B^b_j = b)$ almost coincide with the corresponding simulation results for the entire range of b, with a very slight over estimation for $b = B_{\max,j} = 15$. We also observe

that the analytical results for $P(B_j^g = b)$ generally over estimate 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 under estimation.

Figures 11 through 54 depict the probability that the buffer occupancy is larger than or equal to b, i.e., $P(B_j \geq b) = \sum_{l=b}^{B_{\max,j}} P(B_j = l)$, for different system settings. The plots compare both the theoretical and the simulation results. Figures 11 to 20 show the probability of buffer occupancy of a particular user j, at the end of the runs of good slots for different values of the activity factor a_j and the maximum number of channels that can be used by a client R_j. Each figure has plots for different buffer sizes $B_{\max,j}$. We observe that for b values smaller than $B_{\max,j} - 5$ packets, the buffer occupancy probabilities for different $B_{\max,j}$ lie roughly on one on top of one another. This indicates that the finite buffer analysis gives approximately the same results as the infinite buffer, except when b is close to $B_{\max,j}$. The finite buffer analysis gives smaller $P(B_j > b)$ as we approach the respective $B_{\max,j}$. Thus $Pr(B_j^g = B_{\max,j} | B_{\max,j} = b)$ is less than $Pr(B_j^g \geq b | B_{\max,j} = \infty)$. This can be attributed to the fact that packets that are lost, when the finite buffer is full. In particular, since $q_{good} = 0$ in these examples, the probability of a full buffer at the end of the run of good slots is zero (which is not plotted in the figures). Consider the worst case scenario, where the buffer is full at the end of the run of bad slots. A packet that is generated at the start of the first good slot is dropped and since $q_{good} = 0$, at least one packet is successfully serviced and hence the buffer contents reduce to $B_{\max,j} - 1$.

Figures 21 to 32 depict the buffer occupancy probabilities for different buffer sizes $B_{\max,j}$. Each figure consists of plots corresponding to different activity factors a_j of the client. The probability of buffer occupancy at the end of the bad period, $Pr(B_j^b \geq b)$ is shown in Figures 33 through 54 for various system settings. It can be observed in these plots that the probabilities obtained from analysis tend to flatten out at around $5 \cdot 10^{-6}$. This is because of computational limitations. The maximum number of m and n are chosen such that $\sum_{m,n} \pi_j(m,n) = 0.99999$. Hence the plots saturate above 10^{-6} because we evaluate $Pr(B_j^b = B_{\max,j}) = 1 - \sum_{b_j,b}^{B_{\max,j}-1} Pr(B_j^b = b_{j,b})$. It can be noted from these plots that the simulation results are very close to the analytical results, thus reinforcing the claim that the analysis is quite accurate.

In the following we focus on the performance metrics loss probability and throughput, as defined in Section II-A. In particular,we focus on the actual loss probability $P_l(j) = P(B_j = B_{\max,j})$ as obtained from simulation and the analytical estimates $P(B_j^b = B_{\max,j})$ and $P(B_j^g = B_{\max,j})$; for notational convenience we define $P_l := P_l(j)$, $P_l^b := P(B_j^b = B_{\max,j})$ and $P_l^g := P(B_j^g = B_{\max,j})$ 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 $b = B_{\max,j} = 15$) of the series of points marked "B (S)", "B^b (A)" and "B^g (A)" in Fig 10 respectively. Similarly we consider the actual (simulation) throughput $TH = a_j \cdot [1 - P_l]$ and the analytical estimates $TH^b = a_j \cdot [1 - P_l^b]$ and

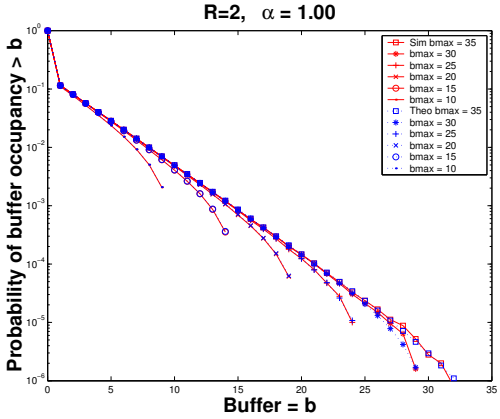

Fig. 11. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 1$ and $R = 2$, $q_{good} = 0$, fixed.

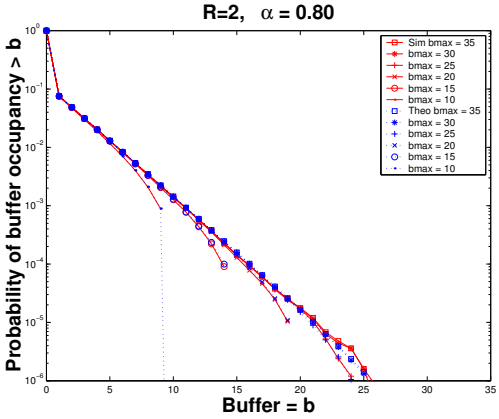

Fig. 12. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.8$ and $R = 2$, $q_{good} = 0$, fixed.

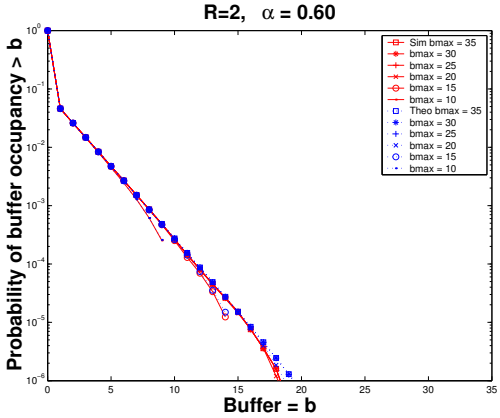

Fig. 13. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.6$ and $R = 2$, $q_{good} = 0$, fixed.

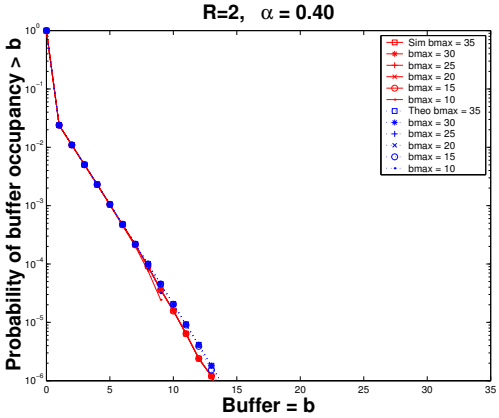

Fig. 14. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.4$ and $R = 2$, $q_{good} = 0$, fixed.

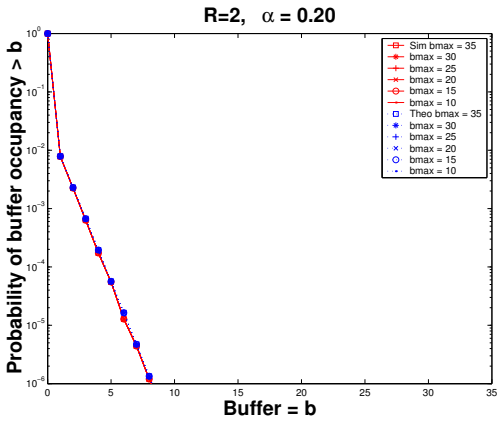

Fig. 15. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.2$ and $R = 2$, $q_{good} = 0$, fixed.

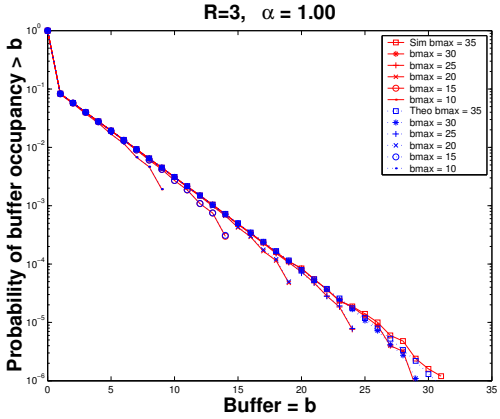

Fig. 16. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 1$ and $R = 3$, $q_{good} = 0$, fixed.

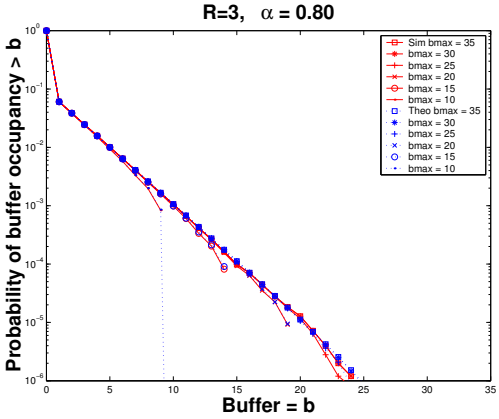

Fig. 17. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.8$ and $R = 3$, $q_{good} = 0$, fixed.

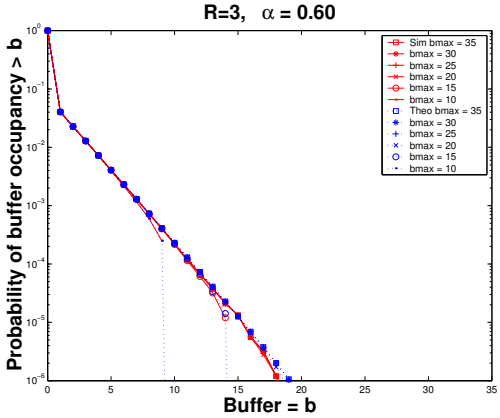

Fig. 18. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.6$ and $R = 3$, $q_{good} = 0$, fixed.

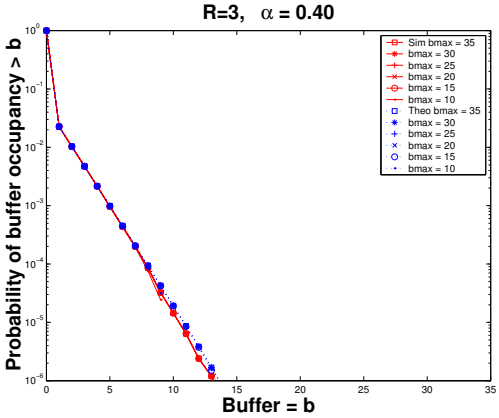

Fig. 19. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.4$ and $R = 3$, $q_{good} = 0$, fixed.

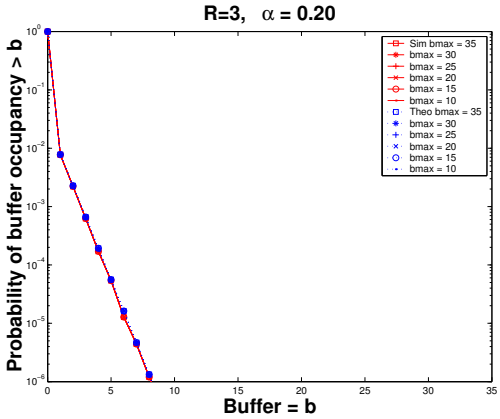

Fig. 20. Probability $P(B_j^g \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.2$ and $R = 2$, $q_{good} = 0$, fixed.

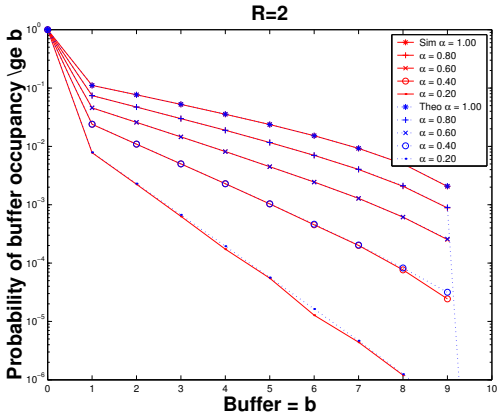

Fig. 21. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 10$ packets, $R = 2$, and $q_{good} = 0$, fixed.

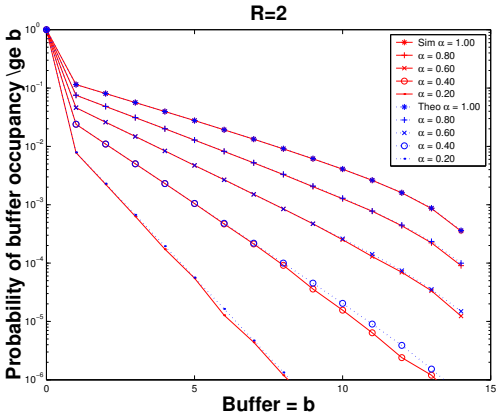

Fig. 22. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 15$ packets, $R = 2$, and $q_{good} = 0$, fixed.

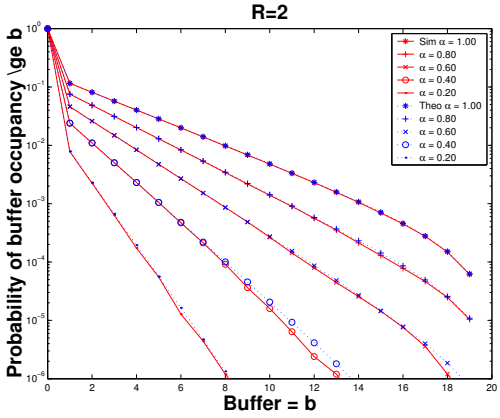

Fig. 23. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 20$ packets, $R = 2$, and $q_{good} = 0$, fixed.

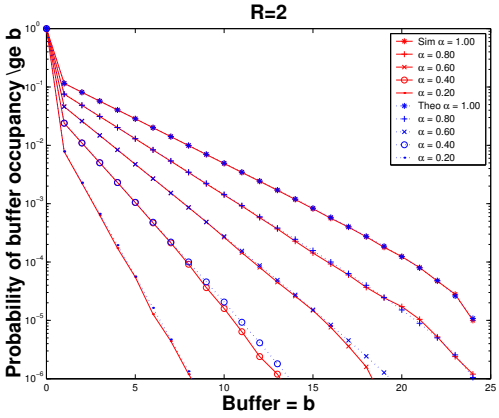

Fig. 24. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 25$ packets, $R = 2$, and $q_{good} = 0$, fixed.

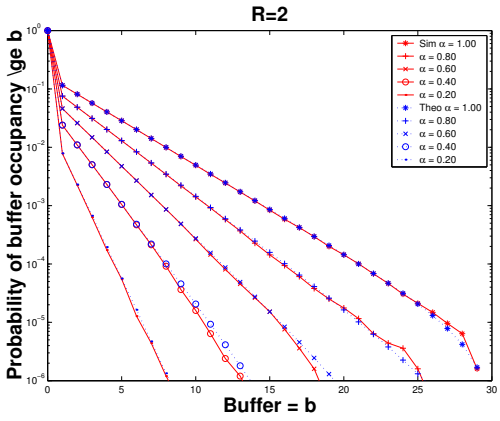

Fig. 25. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 30$ packets, $R = 2$, and $q_{good} = 0$, fixed.

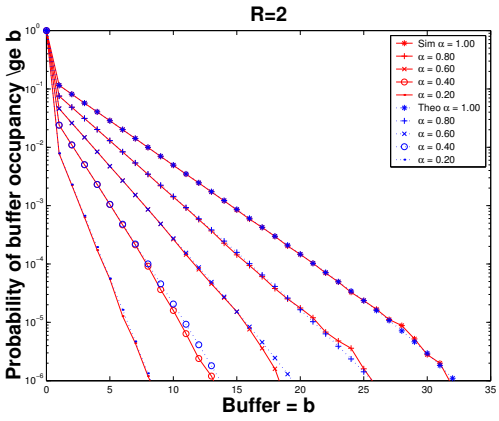

Fig. 26. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 35$ packets, $R = 2$, and $q_{good} = 0$, fixed.

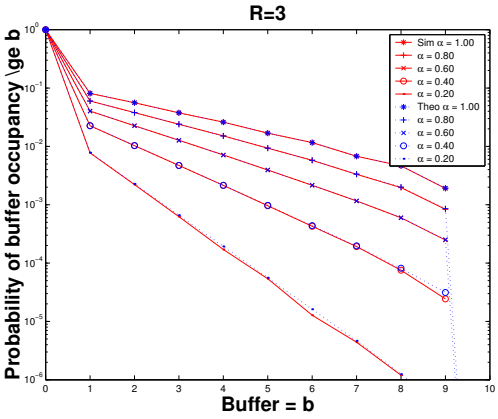

Fig. 27. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 10$ packets, $R = 3$, and $q_{good} = 0$, fixed.

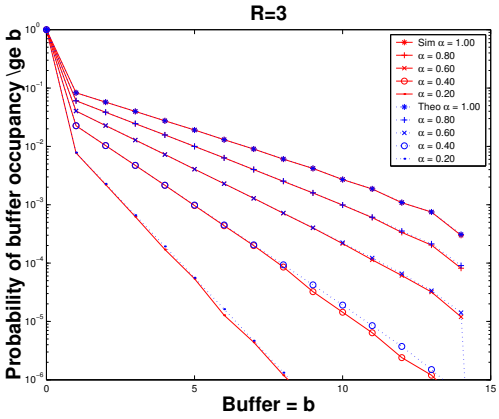

Fig. 28. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 15$ packets, $R = 3$, and $q_{good} = 0$, fixed.

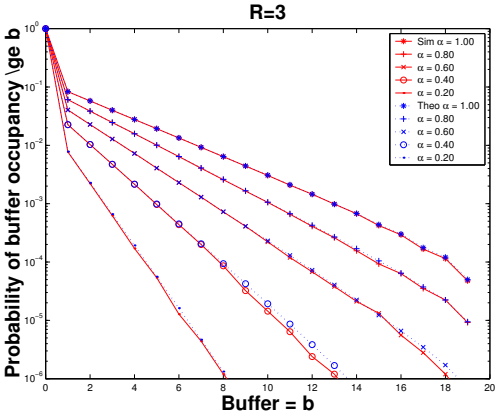

Fig. 29. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 20$ packets, $R = 3$, and $q_{good} = 0$, fixed.

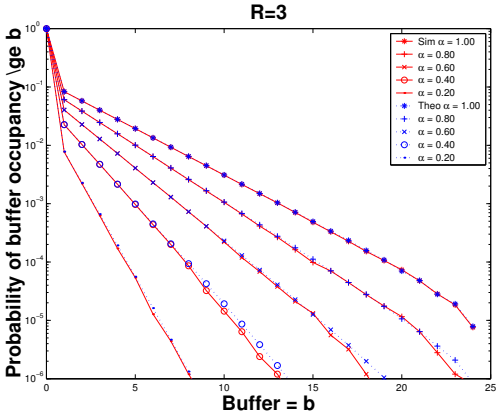

Fig. 30. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 25$ packets, $R = 3$, and $q_{good} = 0$, fixed.

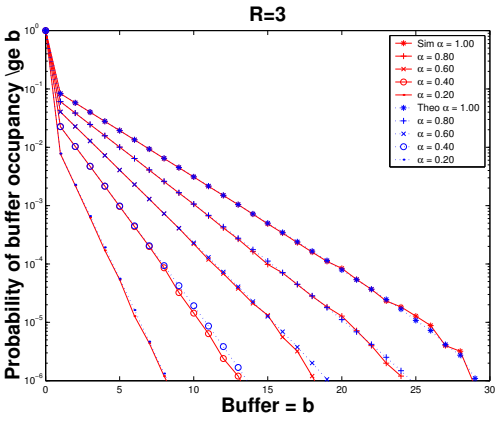

Fig. 31. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 30$ packets, $R = 3$, and $q_{good} = 0$, fixed.

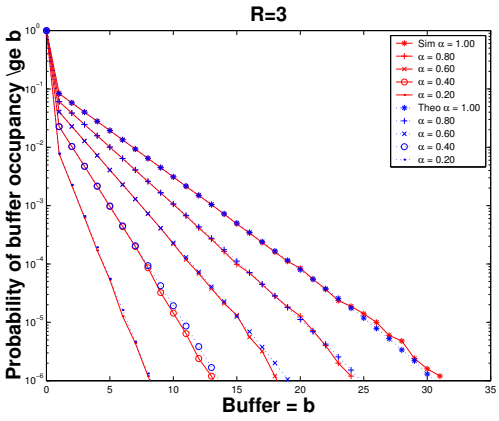

Fig. 32. Probability of $P(B_j^g \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 35$ packets, $R = 3$, and $q_{good} = 0$, fixed.

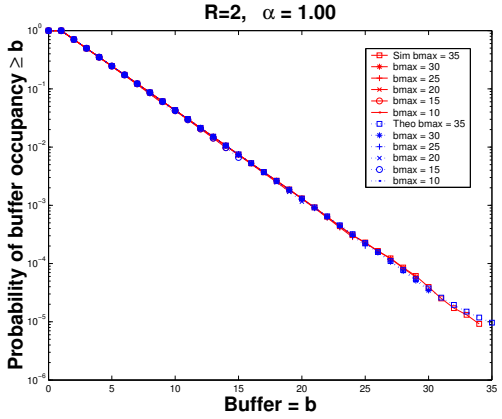

Fig. 33. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 1$ and $R = 2$, $q_{good} = 0$, fixed.

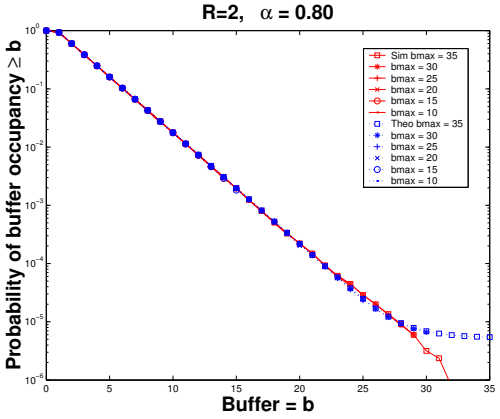

Fig. 34. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.8$ and $R = 2$, $q_{good} = 0$, fixed.

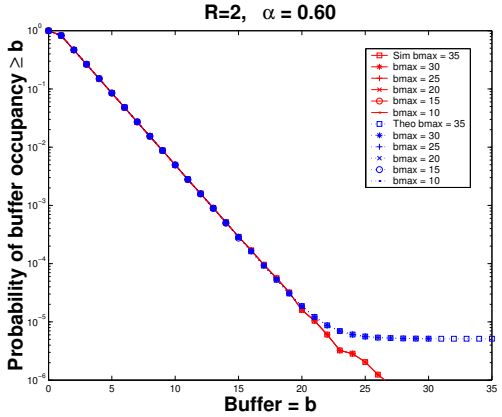

Fig. 35. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.6$ and $R = 2$, $q_{good} = 0$, fixed.

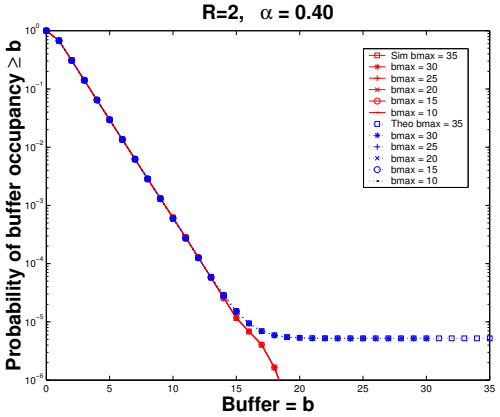

Fig. 36. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.4$ and $R = 2$, $q_{good} = 0$, fixed.

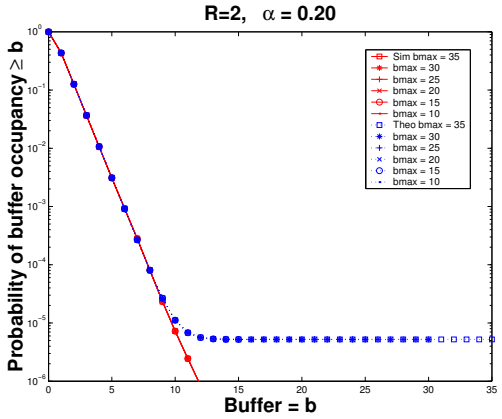

Fig. 37. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.2$ and $R = 2$, $q_{good} = 0$, fixed.

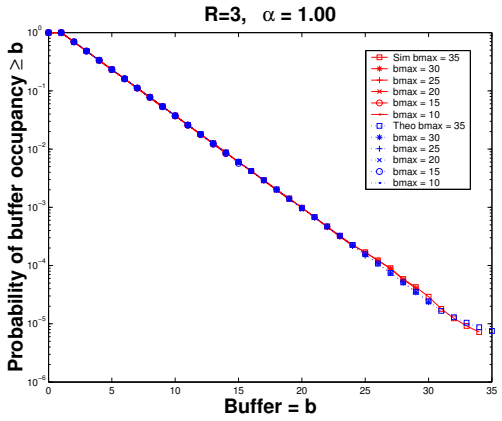

Fig. 38. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 1$ and $R = 3$, $q_{good} = 0$, fixed.

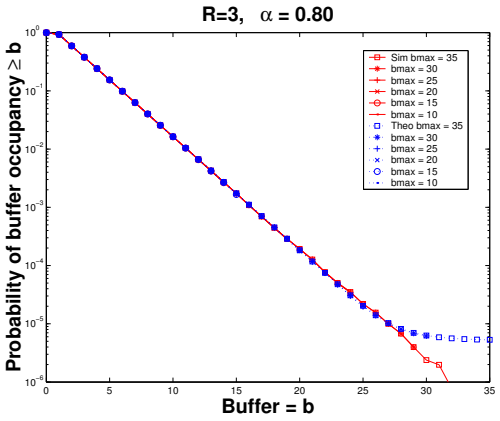

Fig. 39. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.8$ and $R = 3$, $q_{good} = 0$, fixed.

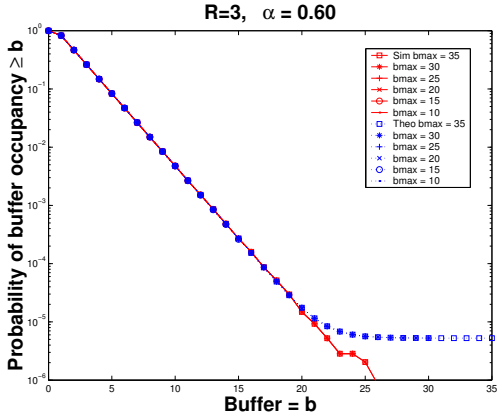

Fig. 40. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.6$ and $R = 3$, $q_{good} = 0$, fixed.

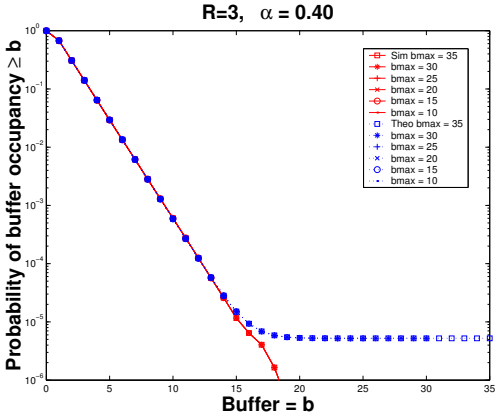

Fig. 41. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.4$ and $R = 3$, $q_{good} = 0$, fixed.

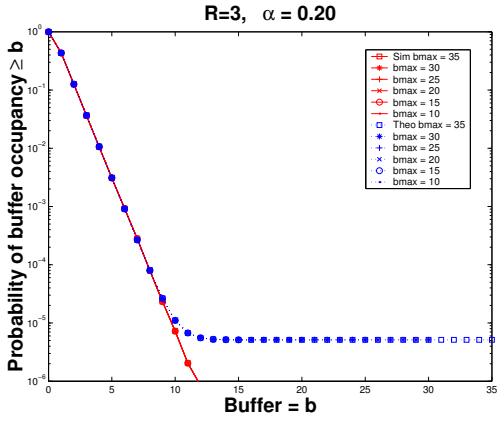

Fig. 42. Probability $P(B_j^b \geq b)$ of an individual user plotted for different values of $B_{\max,j}$, where the activity factor of the user $a_j = 0.2$ and $R = 3$, $q_{good} = 0$, fixed.

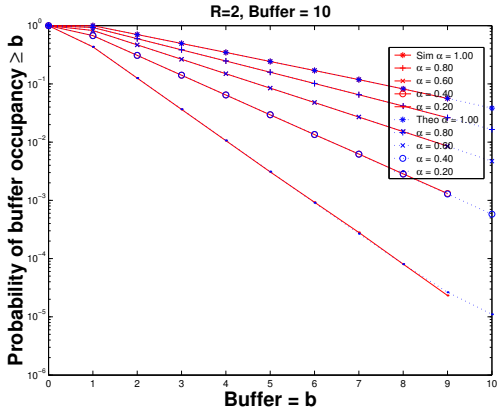

Fig. 43. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 10$ packets, $R = 2$, and $q_{good} = 0$, fixed.

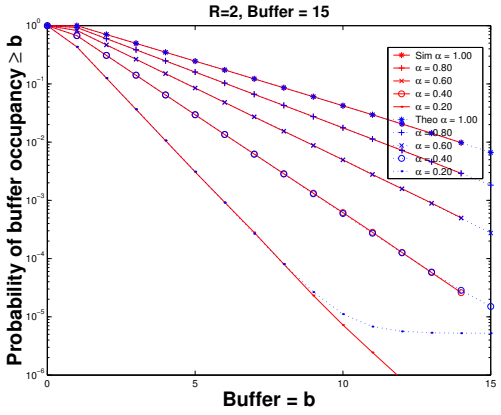

Fig. 44. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 15$ packets, $R = 2$, and $q_{good} = 0$, fixed.

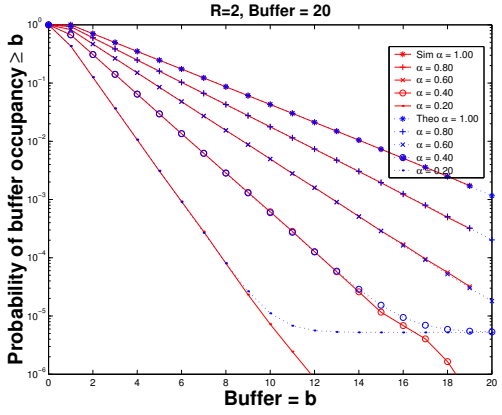

Fig. 45. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 20$ packets, $R = 2$, and $q_{good} = 0$, fixed.

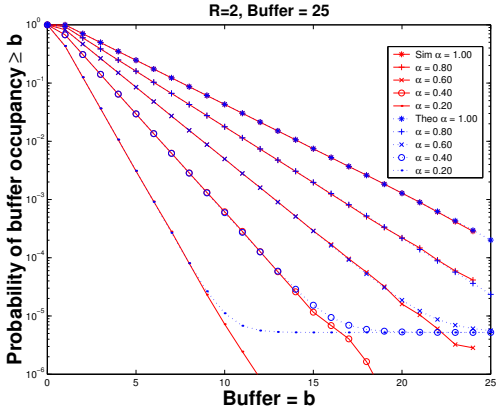

Fig. 46. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 25$ packets, $R = 2$, and $q_{good} = 0$, fixed.

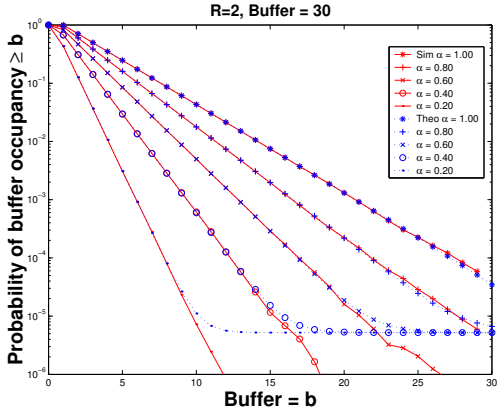

Fig. 47. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 30$ packets, $R = 2$, and $q_{good} = 0$, fixed.

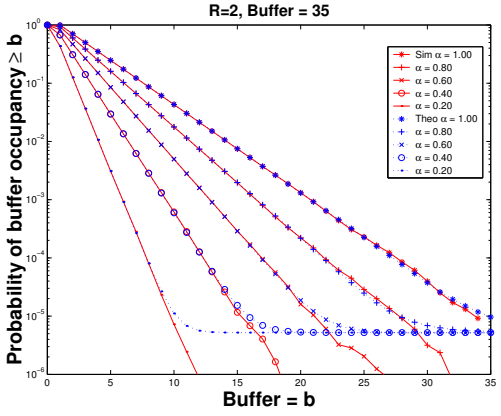

Fig. 48. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 35$ packets, $R = 2$, and $q_{good} = 0$, fixed.

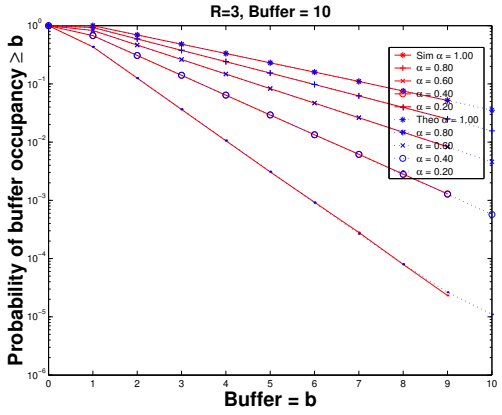

Fig. 49. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 10$ packets, $R = 3$, and $q_{good} = 0$, fixed.

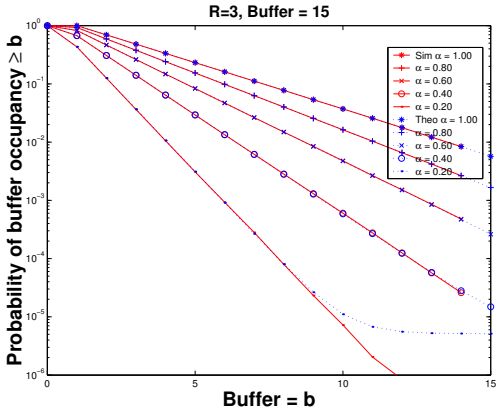

Fig. 50. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 15$ packets, $R = 3$, and $q_{good} = 0$, fixed.

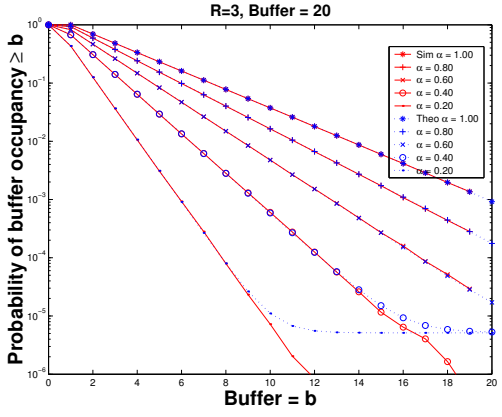

Fig. 51. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 20$ packets, $R = 3$, and $q_{good} = 0$, fixed.

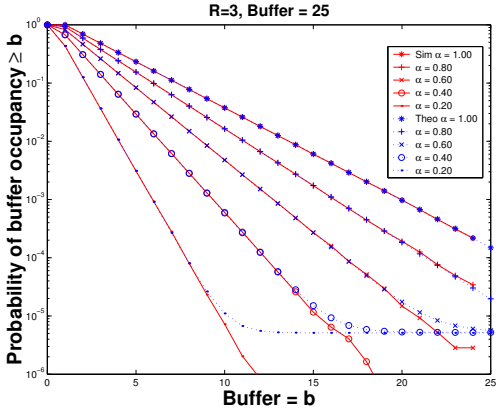

Fig. 52. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 25$ packets, $R = 3$, and $q_{good} = 0$, fixed.

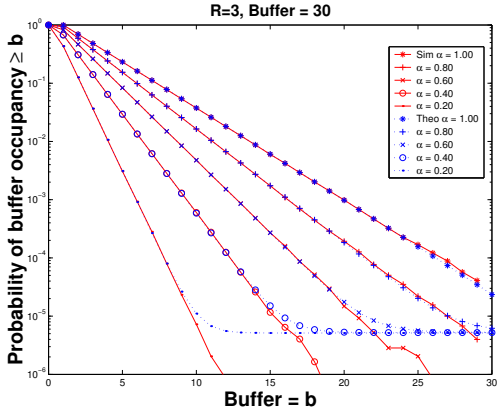

Fig. 53. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 30$ packets, $R = 3$, and $q_{good} = 0$, fixed.

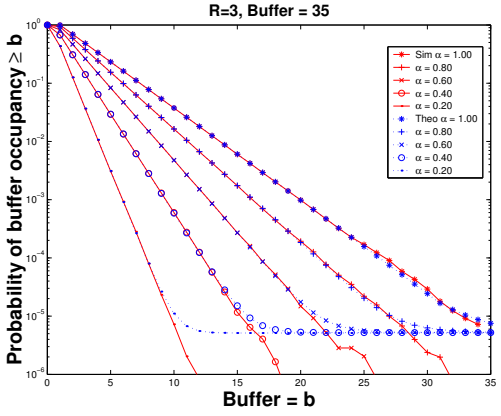

Fig. 54. Probability of $P(B_j^b \geq b)$ of an individual user plotted for different activity factors a_j, where $B_{\max,j} = 35$ packets, $R = 3$, and $q_{good} = 0$, fixed.

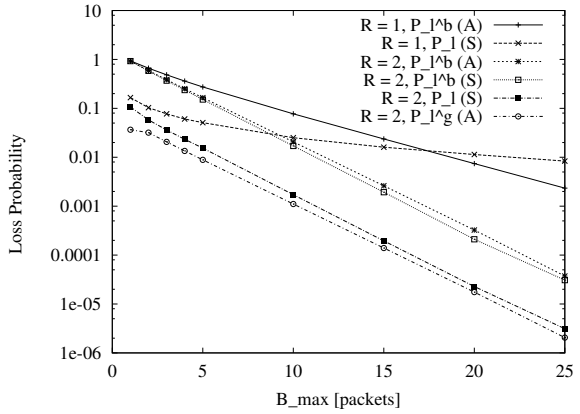

Fig. 55. Loss Probability as a function of link layer buffer capacity $B_{\max,j}$ for conventional ARQ ($R_j = 1$) and SMPT with $R_j = 2$ ($a_j = 0.8$, $q^j_{good} = 0.02$, fixed).

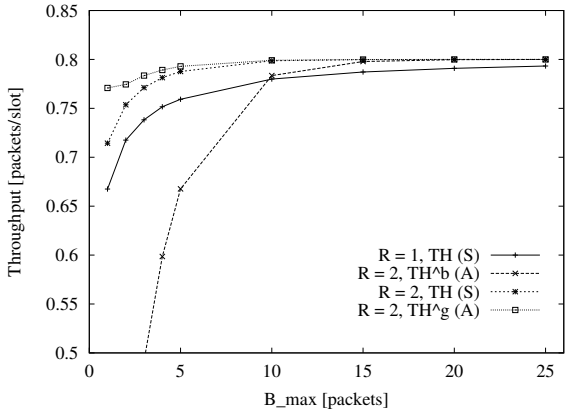

Fig. 56. Throughput of a client as a function of buffer capacity $B_{\max,j}$ for conventional ARQ ($R_j = 1$) and SMPT with $R_j = 2$ ($a_j = 0.8$, $q^j_{good} = 0.02$, fixed).

$TH^g = a_j \cdot [1 - P_l^g]$.

In Figure 55 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 $B_{\max,j}$ in packets. Similarly, in Figure 56 we plot the throughput TH (obtained from simulation) and the analytical estimates TH^b and TH^g (all in packet per slot) as a function of $B_{\max,j}$ in packets. For both figures, the activity factor is $a_j = 0.8$ and the packet drop probability in the good state is $q_{good}(j) = 0.02$. We provide results for a maximum number of $R_j = 1$ and $R_j = 2$ usable CDMA code channels. We observe from Figure 55 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 under estimates 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 $R_j = 2$.

For $R_j = 1$, our analytical results are not quite as good. We observe from Figure 55 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 Figure 56 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 sub-slots of a wireless CDMA system running some form of SMPT. We consider the independent link model throughout this section. 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 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

$$P(C_j(a_j) = c), \;\; c = 0, \ldots, R_j.$$

Initially, we analyze the channel usage for an activity factor of $a_j = 1$, subsequently we extend 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,

$$P\left(C_j(1) = c\right) = \frac{\sum_{b=0}^{B_{\max,j}} \sum_{m,n>0} \mathcal{N}_c(m, n, b) \cdot \pi_j(m, n) \cdot Pr(B_j^g = b)}{\overline{T_j}}, \quad c \;=\; 0, 1, \ldots R_j. \tag{21}$$

We now proceed to calculate $\mathcal{N}_c(m, n, b)$ for the different forms of SMPT described in Section III for $c = 2, \ldots, R_j$; 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

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

Case I: $2 \le c < R_j$

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

Subcase I.2: $b \le c-2$ and $m+b \ge R_j$

Subcase I.3: $\text{mod}(m+b,\ R-1) = c-1$ and $n \ge \left\lfloor \frac{m+b}{R-1} \right\rfloor + 1$

$$\mathcal{N}_c(m,n,b) = \begin{cases} 2 & \text{if Subcases I.1 and I.3 hold,} \\ 2 & \text{if Subcases I.2 and I.3 hold,} \\ 1 & \text{if one of the Subcases I.1, I.2 or I.3 holds,} \\ 0 & \text{otherwise.} \end{cases}$$

Case II: $c = R_j$

Subcase II.1: $b \ge R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor \ge n \Rightarrow \mathcal{N}_c(m,n,b) = m+n$

Subcase II.2: $b < R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor \ge n \Rightarrow \mathcal{N}_c(m,n,b) = m - R_j + b + 1 + n$

Subcase II.3: $b \ge R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor < n \Rightarrow \mathcal{N}_c(m,n,b) = m + \left\lfloor \frac{m+b}{R-1} \right\rfloor$

Subcase II.4: $b < R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor < n \Rightarrow \mathcal{N}_c(m,n,b) = m - R_j + b + 1 + \left\lfloor \frac{m+b}{R-1} \right\rfloor$

Defn.:

$\theta(m,b) = \text{mod}(m+b,\ R_j - 1)$

$\eta(m,b) = \left\lfloor \frac{m+b}{R_j - 1} \right\rfloor$

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

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

Summary:

$$\mathcal{N}_c(m,n,b) = \begin{cases} \delta(b - R_j - 1) + u(c-2-b) \cdot u(m+b-c) + \delta(\theta(m,b) - c + 1) \cdot u(n - \eta(m,b) - 1) \\ \qquad \text{if } 2 \le c < R_j \\ \max(m - \max(R_j - 1 - b, 0),\ 0) + \min(\eta(m,b),\ n) \\ \qquad \text{otherwise.} \end{cases}$$

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 $2 \le c < R_j$, there are up to 2 possibilities to use c codes: (i) 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 (ii) when $c-1$ backlogged packets are left when clearing the backlog (Subcase I.3). For $c = R_j$, we need to distinguish the scenarios, where R_j codes are used towards 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 where $R_j - 1$ backlogged packets are cleared in each of the good slots (Subcases II.1 and II.2) or where less 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 fast healing SMPT the analysis is very similar to the outlined analysis for basic SMPT and we refer to the Appendix for a summary of the analysis and the expression for $\mathcal{N}_c(m, n, b)$ for fast healing SMPT. 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 the Appendix for details. With the derived $\mathcal{N}_c(m, n, b),\ \ c = 2, \ldots, R_j$, we evaluate $P(C_j(1) = c),\ c = 2, \ldots, R_j$ using (21), 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 $P(C_j(a_j) = c) = a_j \cdot P(C_j(1) = c),\ c = 1, 2, \ldots, R_j$.

B. Channel Usage (Interference Level) in 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. We consider 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, $j = 1, \ldots, J$, is given by $P(C_j(a_j) = c)$, $c = 0, \ldots, R_j$, provided in the previous section.

Let $j^\star$ denote the considered client and note that the transmissions of the other clients $j = 1, \ldots, J,\ j \neq j^\star$ are interfering with the transmissions of client $j^\star$. Let C be a discrete random variable denoting the total number of used CDMA codes by the interfering clients $j = 1, \ldots, J,\ j \neq j^\star$ in a given uplink subslot in steady-state. Note that C gives the interference level for client $j^\star$ (in number of used codes) in the cell. With the independent link model the distribution $P(C = c)$, $c = 0, \ldots, \sum_{j \neq j^\star} R_j = R_{tot}$, is given by the convolution of the individual distributions $P(C(a_j) = c)$, $j = 1, \ldots, J,\ j \neq j^\star$. For notational convenience we define $\overline{C}$ as the average total number of interfering codes, i.e., $\overline{C} = \sum_{c=0}^{R_{tot}} c \cdot P(C = c)$. Also, let σ_C^2

denote the variance of the random variable C, i.e., $\sigma_C^2 = \sum_{c=0}^{R_{tot}} (c - \overline{C})^2 \cdot P(C = c)$.

VI. Analysis of Buffer Occupancy at wireless client with Interference Link Model

In this section we analyze the buffer occupancy of a wireless client $j^\star$ with the interference link model. This analysis takes the interference in the cell due to the uplink transmissions of all interfering clients, $j = 1, \ldots, J,\ j \neq j^\star$ 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 $q_{bit}(\overline{C} + \tau \cdot \sigma_C)$ using (8), where τ, $\tau > 0$, denotes the tuning parameter. We then evaluate the packet drop probability in the good state of the interference link model as $q_{good}(\overline{C} + \tau \cdot \sigma_C)$ using (11). Note that we approximate the number of interfering channels in a CDMA system running SMPT by $\overline{C} + \tau \cdot \sigma_C$. 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) indicates that this approximation has good accuracy for $\tau = 1$. The packet drop probability $q_{good}(\overline{C} + \tau \cdot \sigma_C)$ 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 multi-code 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 ($a_j = a\ \forall j = 1, \ldots, J$) and maximum number of usable codes ($R_j = R,\ \forall j = 1, \ldots, J$). 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 $P_l^b := \frac{1}{J} \sum_{j=1}^{J} P(B_j^b = B_{\max,j})$ and $P_l^g := \frac{1}{J} \sum_{j=1}^{J} P(B_j^g = B_{\max,j})$. We also consider the corresponding aggregate throughput TH as defined in (3) and obtained from simulation, as well as the analytical estimates $TH^b = \sum_{j=1}^{J} a_j \cdot (1 - P(B_j^b = B_{\max,j}))$ and $TH^g = \sum_{j=1}^{J} a_j \cdot (1 - P(B_j^g = B_{\max,j}))$. We also consider the average link layer buffer occupancy in the clients $B_{\text{avg}} = \frac{1}{J} \sum_{j=1}^{J} \sum_{b=1}^{B_{\max,j}} b \cdot P(B_j = b)$ and the standard deviation of the link layer buffer occupancy B_{std}. In Figures 57 through 80 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 with $B_{\max} = 20$ packets, in the cell for different combinations of the activity factor a and the number of usable codes R. In Figures 83 through 94 we compare the average loss probability and aggregate throughput of systems employing conventional ARQ and systems employing SMPT with $R = 2$, as a function of the number of clients J with $B_{\max} = 5$ packets, in the cell for different activity factors a.

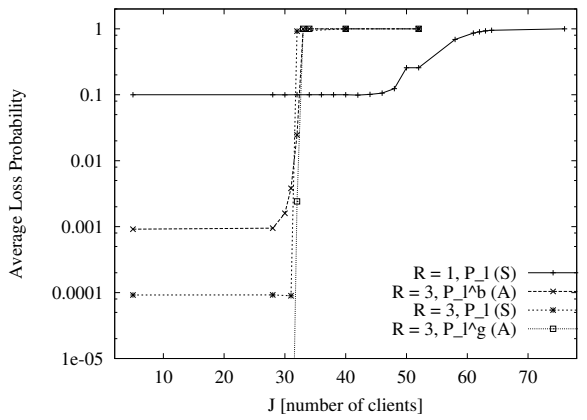

Fig. 57. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 1$, $B_{\max} = 20$ packets, fixed).

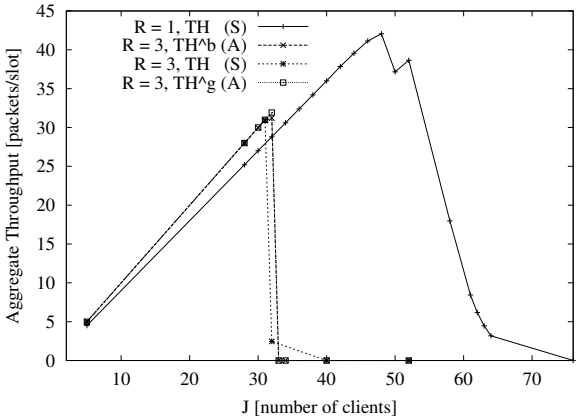

Fig. 58. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 1$, $B_{\max} = 20$ packets, fixed).

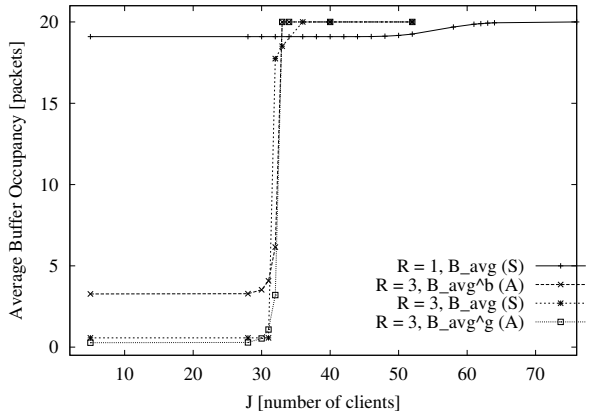

Fig. 59. Average buffer occupancy B_{avg} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 1$,$B_{\max} = 20$ packets, fixed).

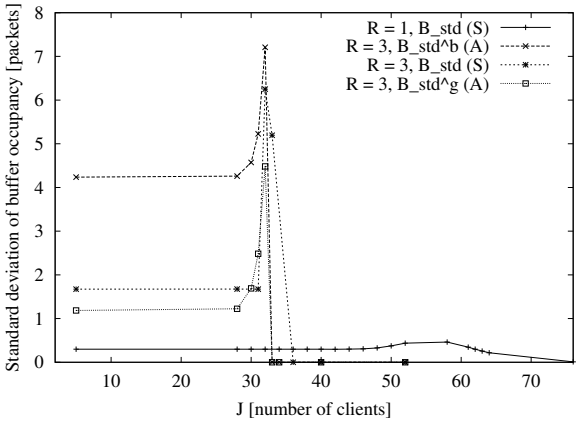

Fig. 60. Standard deviation of buffer occupancy B_{std} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 1$, $B_{\max} = 20$ packets, fixed).

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 $R = 1$ and $B_{\max} = 20$. This difference is most pronounced for an activity factor of $a = 1$. With conventional ARQ, 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 arriving packets.) With a client's channel being in the bad state with a probability of $P_{bad}^{j} = \alpha/(\alpha + \beta) = 0.1$ (in the considered scenario with $\alpha = 1/27$ and $\beta = 1/3$) and a new packet arriving in every slot (with activity factor $a = 1$)

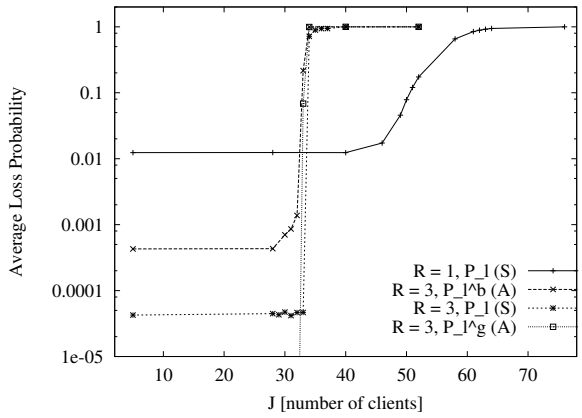

Fig. 61. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 0.9$, $B_{\max} = 20$ packets, fixed).

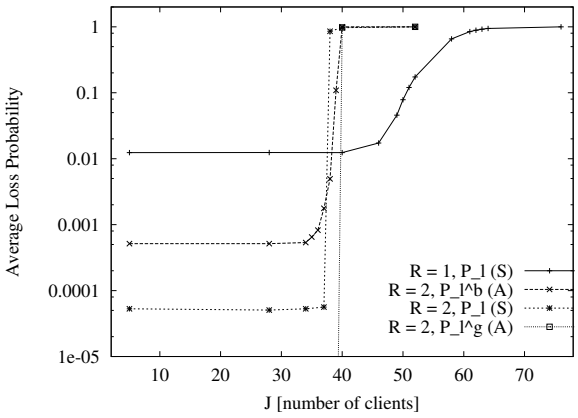

Fig. 62. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.9$, $B_{\max} = 20$ packets, fixed).

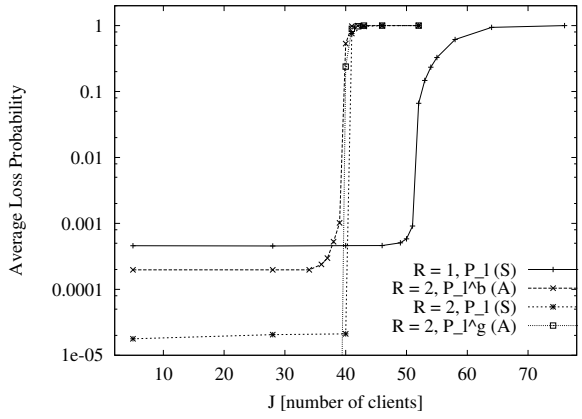

Fig. 63. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.8$, $B_{\max} = 20$ packets, fixed).

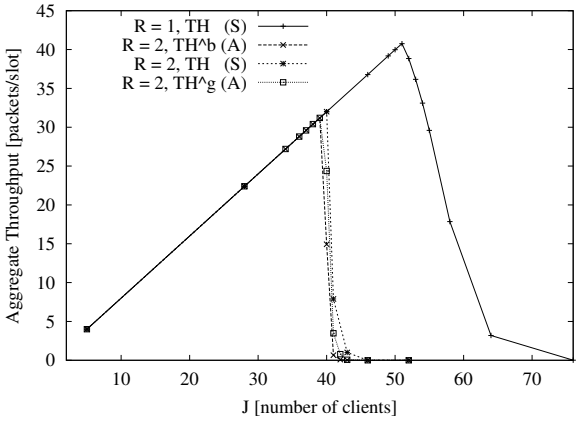

Fig. 64. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.8$, $B_{\max} = 20$ packets, fixed).

this results in an average loss probability of roughly 0.1 for low interference levels (small J). With SMPT the average loss probability for $a = 1$ 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, For $a = 0.6$, the loss probability with SMPT is just about half order of magnitude smaller than the loss probability with conventional ARQ. This indicates that as the activity factor decreases, the advantages of SMPT with respect to conventional ARQ in the average packet loss gradually diminish. As the activity factor is reduced further, say to, $a = 0.4$ in this scenario with $B_{\max} = 20$, no loss

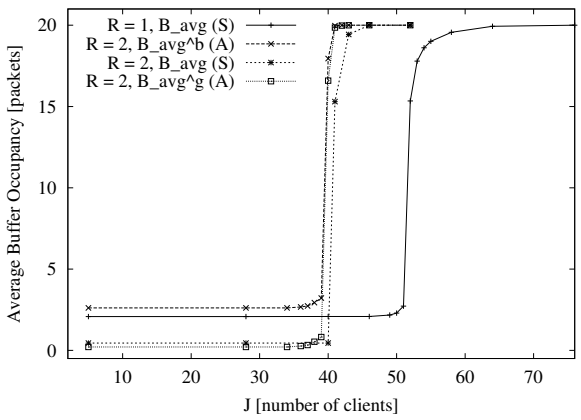

Fig. 65. Average buffer occupancy B_{avg} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.8$, $B_{\max} = 20$ packets, fixed).

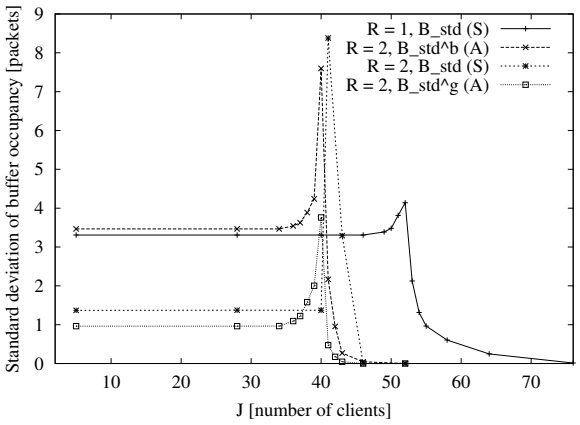

Fig. 66. Standard deviation of buffer occupancy B_{std} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.8$, $B_{\max} = 20$ packets, fixed).

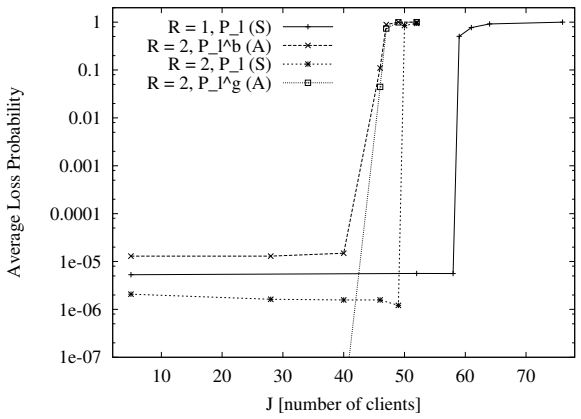

Fig. 67. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

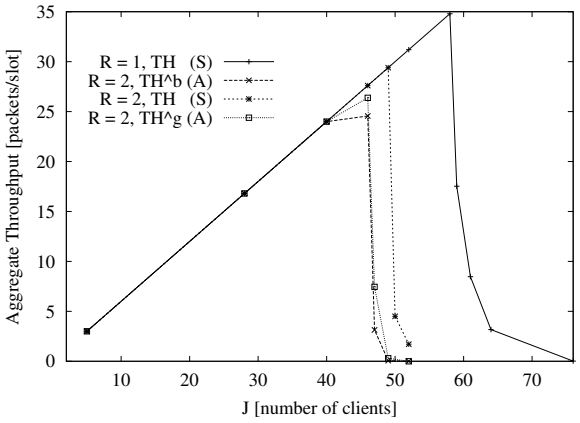

Fig. 68. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

was observed in very long simulation runs and the analysis gives a loss probability of zero as depicted in Figure 79. For this low activity scenario, SMPT does not provide any advantage over conventional ARQ. In fact, the system capacity (number of supported flows) is slightly smaller with SMPT than with conventional ARQ, for the same no-loss QoS. We note, however, that in a typical wireless communications scenario, long upper layer IP packets are segmented into several small link (MAC) layer packets for transmission over the wireless link. In this typical scenario, the traffic process at the link layer consists of periods of activity (MAC layer packets of an IP packets are being served) and inactivity (silent periods in between successive IP packets). During the periods of activity the link layer "sees" an activity factor of one, i.e., a new MAC

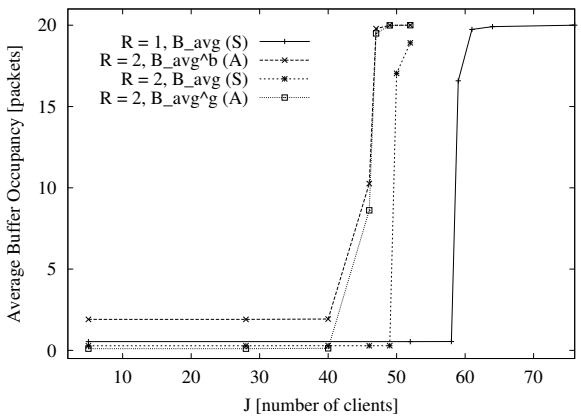

Fig. 69. Average buffer occupancy B_{avg} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

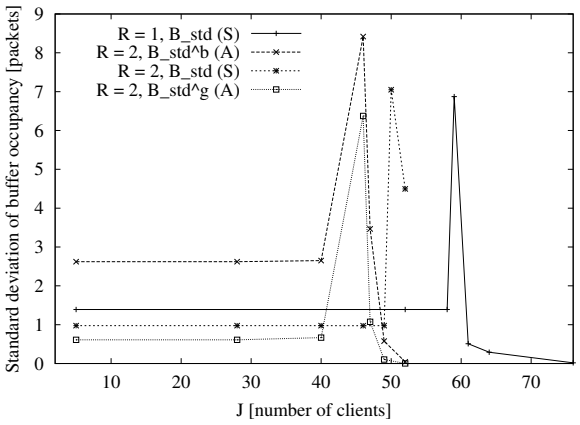

Fig. 70. Standard deviation of buffer occupancy B_{std} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

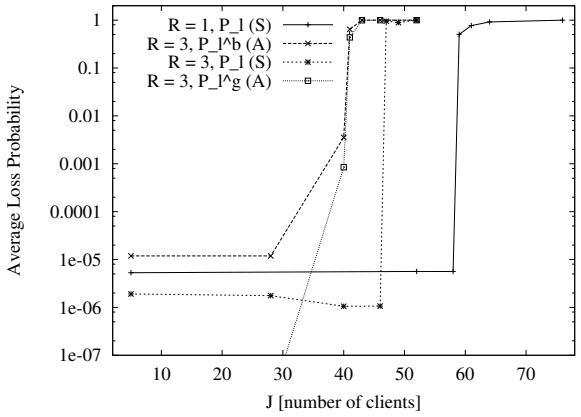

Fig. 71. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

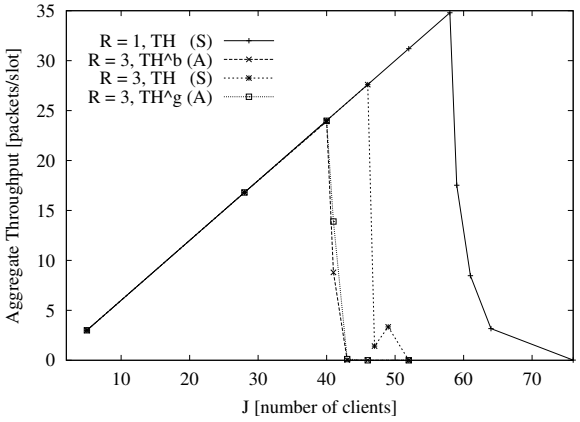

Fig. 72. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

packet is to be served in each slot, whereas the activity factor is zero during periods of inactivity. Our model with independent Bernoulli packet arrivals with $a_j = 1$ in each slot is a first approximation of the system behavior when all clients' activity periods collude and the activity periods are long. (In future work we plan to extend our traffic model to Markov modulated Bernoulli MAC packet arrivals, which capture the effect of periods of activity and inactivity at the MAC layer.) As our simulation and analytical results indicate, for an activity factor of one, SMPT gives significant improvements in link layer QoS at the expense of a moderate reduction of capacity. With a buffer capacity of $B_{\max} = 5$ packets in each client's link layer, SMPT with $R = 2$ achieves a reduction of the loss probability of roughly one order of magnitude for $a_j = 1$ compared

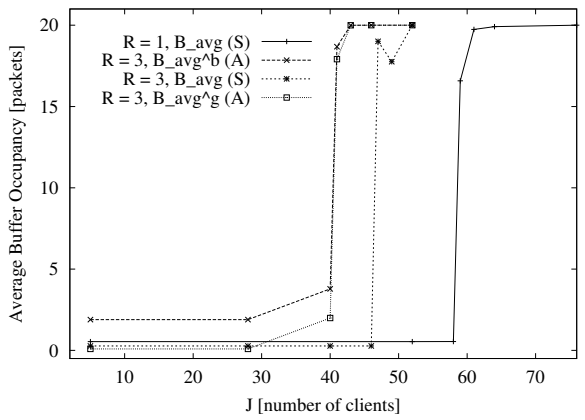

Fig. 73. Average buffer occupancy B_{avg} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

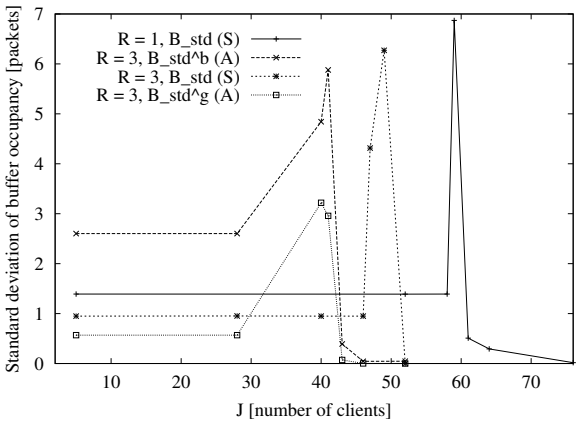

Fig. 74. Standard deviation of buffer occupancy B_{std} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 3$ (activity factor $a = 0.6$, $B_{\max} = 20$ packets, fixed).

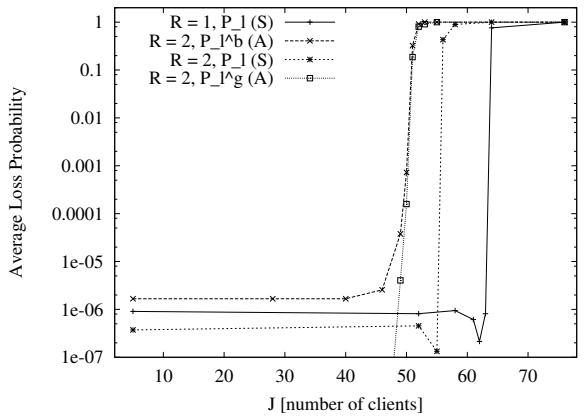

Fig. 75. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.5$, $B_{\max} = 20$ packets, fixed).

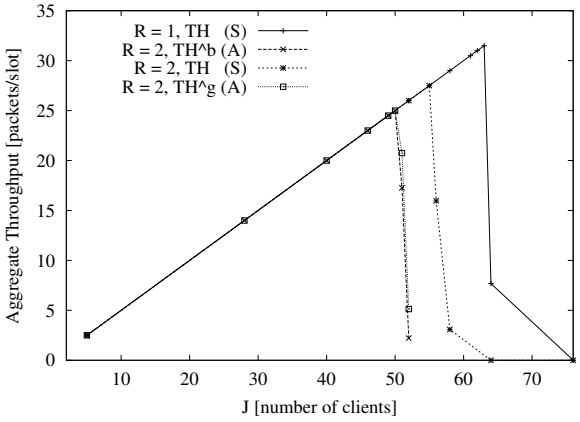

Fig. 76. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.5$, $B_{\max} = 20$ packets, fixed).

to conventional ARQ. SMPT supports about 24% fewer flows at this higher level of QoS than conventional ARQ. We also note that for smaller activity factors, which are of limited relevance in practice, the QoS improvement achieved with SMPT decreases.

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 Figure 55 with the results in Figure 63, note that Figure 55 gives results for a fixed $q_{good} = 0.02$, whereas for Figure 63 q_{good} is a function of the interference level, i.e., the

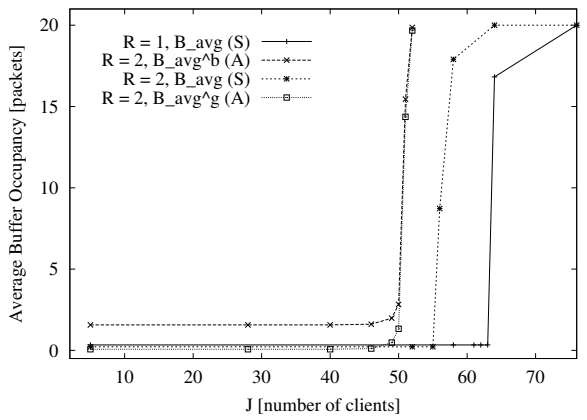

Fig. 77. Average buffer occupancy B_{avg} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.5$, $B_{\max} = 20$ packets, fixed).

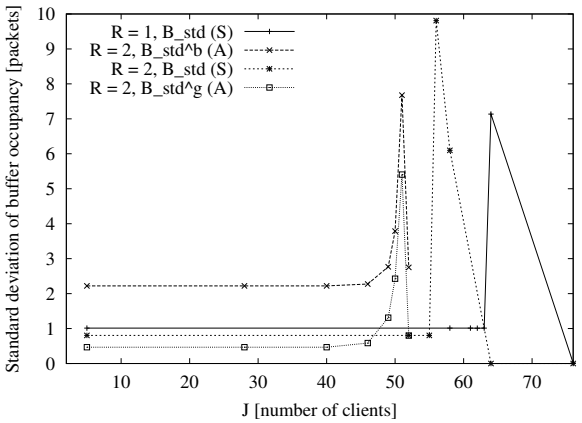

Fig. 78. Standard deviation of buffer occupancy B_{std} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.5$, $B_{\max} = 20$ packets, fixed).

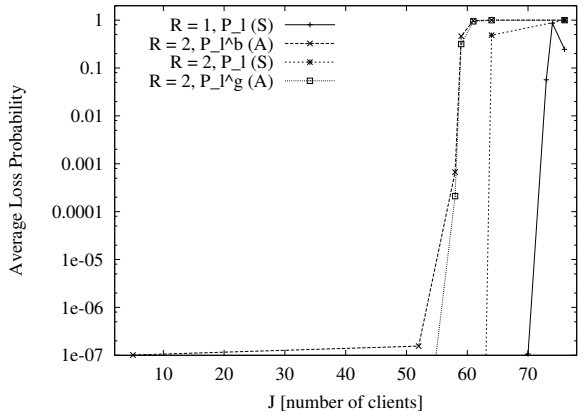

Fig. 79. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.4$, $B_{\max} = 20$ packets, fixed).

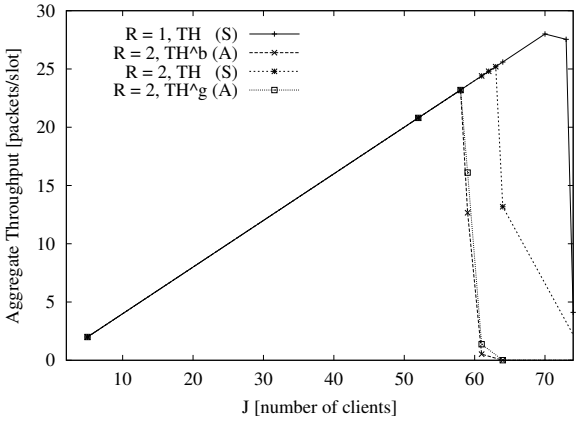

Fig. 80. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.4$, $B_{\max} = 20$ packets, fixed).

number of clients J, as described in Section II-B. For small J, q_{good} is essentially zero in the interference link model, as can be seen from Figure 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 which 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 fol-

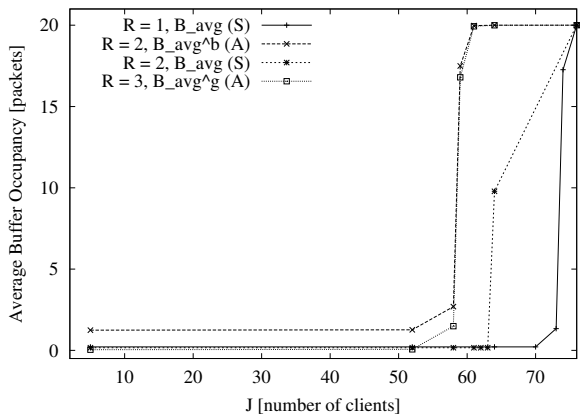

Fig. 81. Average buffer occupancy B_{avg} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.4$, $B_{\max} = 20$ packets, fixed).

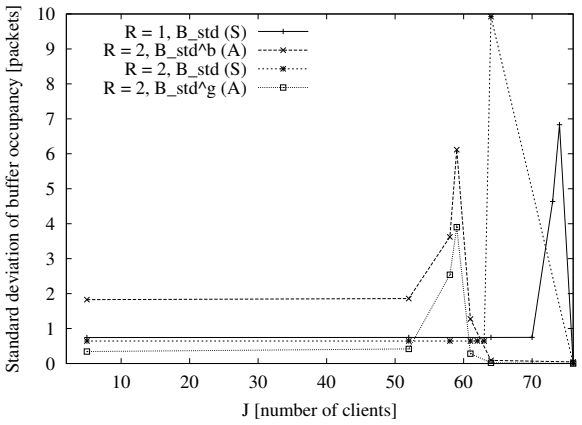

Fig. 82. Standard deviation of buffer occupancy B_{std} as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.4$, $B_{\max} = 20$ packets, fixed).

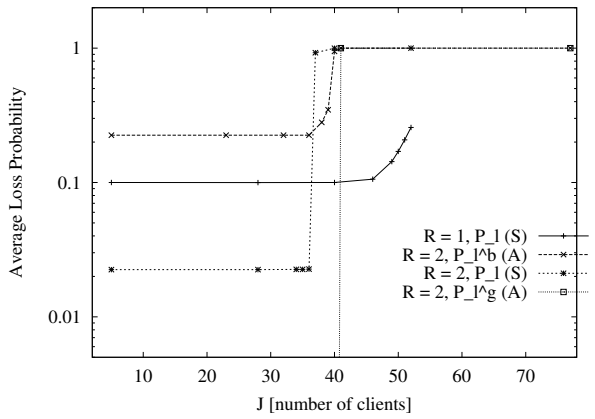

Fig. 83. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 1$, $B_{\max} = 5$ packets, fixed).

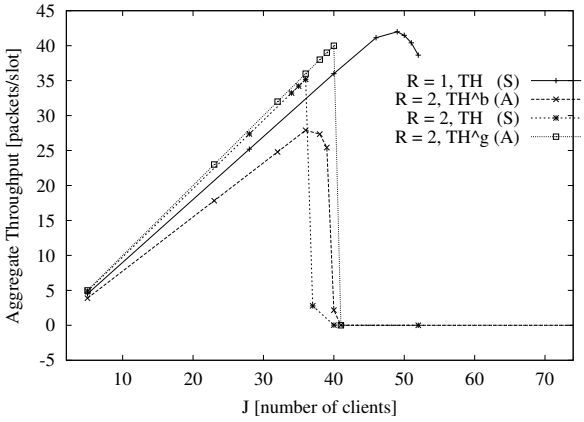

Fig. 84. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 1$, $B_{\max} = 5$ packets, fixed).

lows. 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 Figure 58 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. 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

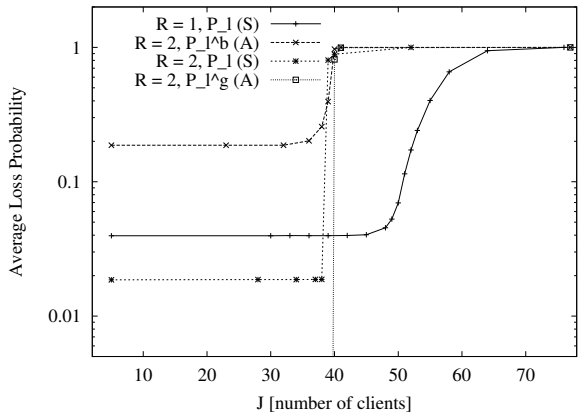

Fig. 85. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.9$, $B_{\max} = 5$ packets, fixed).

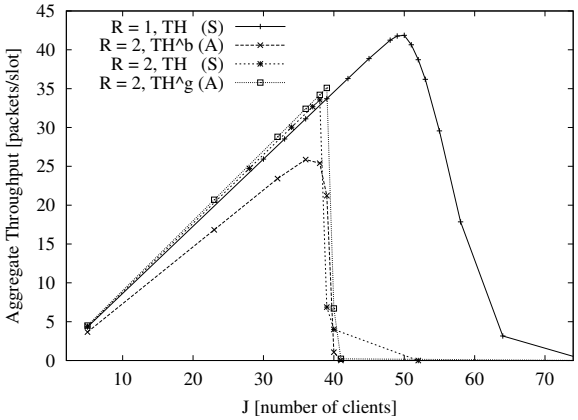

Fig. 86. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.9$, $B_{\max} = 5$ packets, fixed).

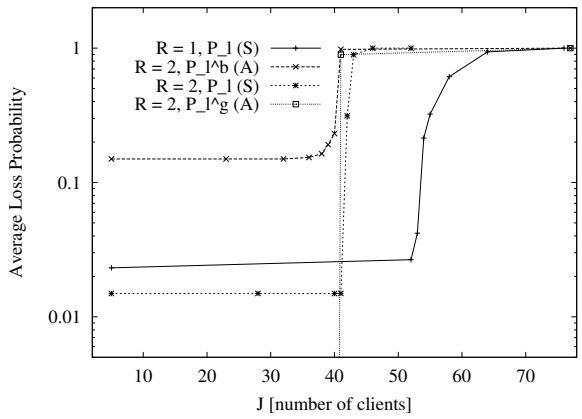

Fig. 87. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.8$, $B_{\max} = 5$ packets, fixed).

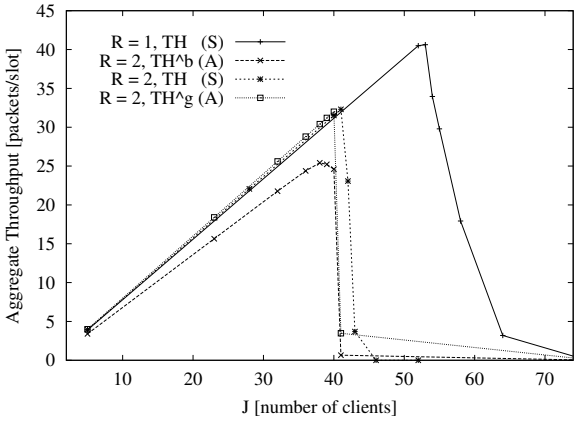

Fig. 88. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.8$, $B_{\max} = 5$ packets, fixed).

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 Figure 62 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

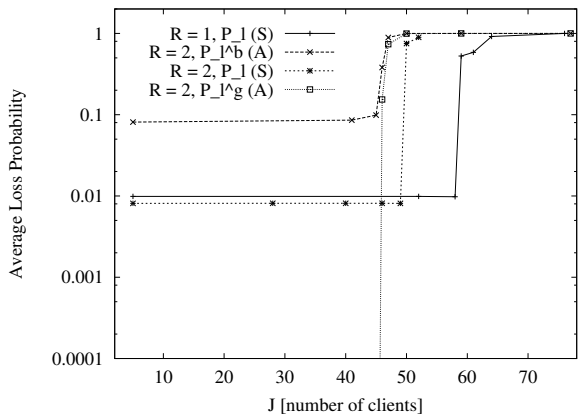

Fig. 89. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.6$, $B_{\max} = 5$ packets, fixed).

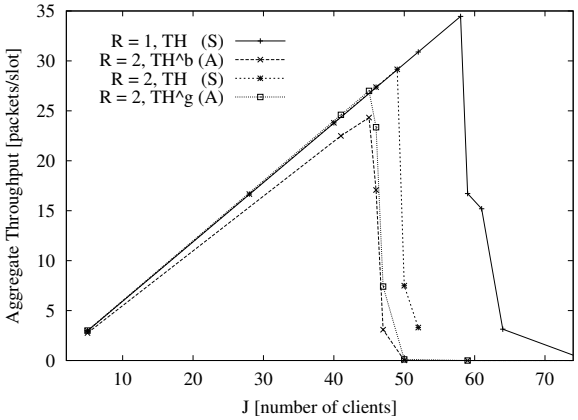

Fig. 90. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.6$, $B_{\max} = 5$ packets, fixed).

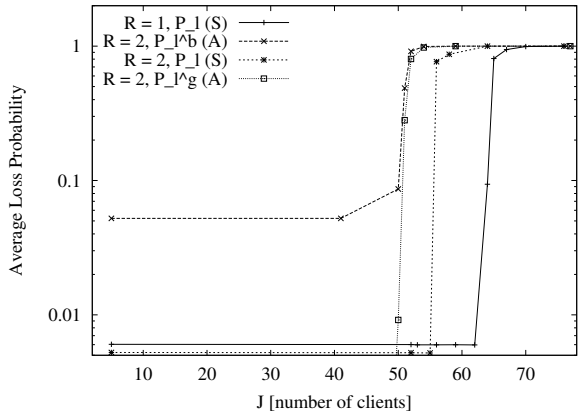

Fig. 91. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.5$, $B_{\max} = 5$ packets, fixed).

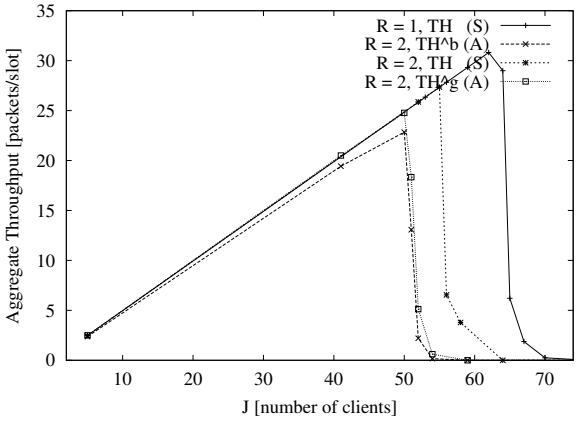

Fig. 92. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.5$, $B_{\max} = 5$ packets, fixed).

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 Figure 59 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 msec this translates into a delay of (at least) 190 msec. 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 Figure 60 is due to the buffer typically holding either 19 or

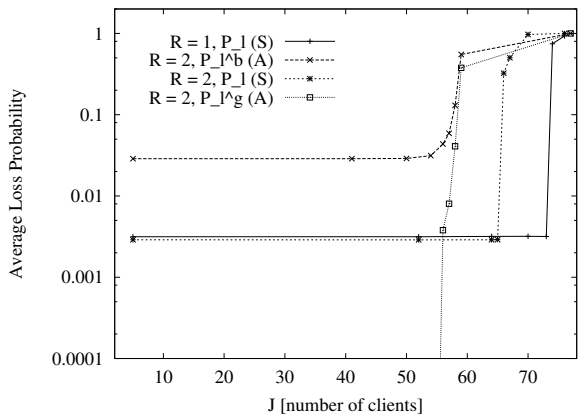

Fig. 93. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.4$, $B_{\max} = 5$ packets, fixed).

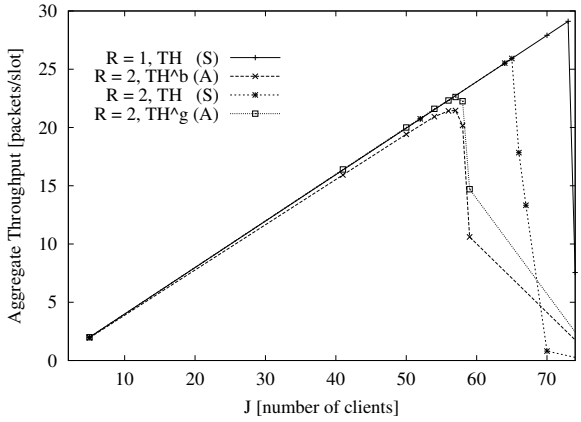

Fig. 94. Aggregate throughput in cell as a function of number of supported clients J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (activity factor $a = 0.4$, $B_{\max} = 5$ packets, fixed).

20 packets and never being drained to lower levels. 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 Figure 59. We observe from Figure 65 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 Figure 66 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 UDP, some of these applications are likely to run over TCP, e.g., see [78], [79], [3]. The large link layer losses as well as the large delays and delay jitters of conventional ARQ tend to trigger frequent TCP re-transmissions and cuts in 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 [80], [81]. 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 Figure 61 and 62 that SMPT with $R = 3$ gives very slightly smaller loss probabilities and a slightly smaller capacity than

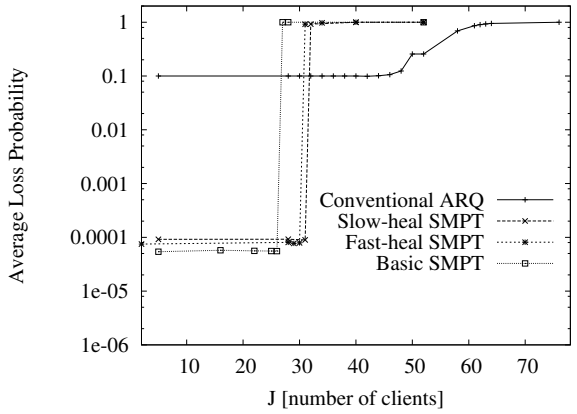

Fig. 95. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ, slow-healing SMPT, fast-healing SMPT, and basic SMPT with $R = 3$ (non-bursty traffic with activity factor $a = 1.0$, the arrivals being Bernoulli process, $B_{\max} = 20$ packets, fixed).

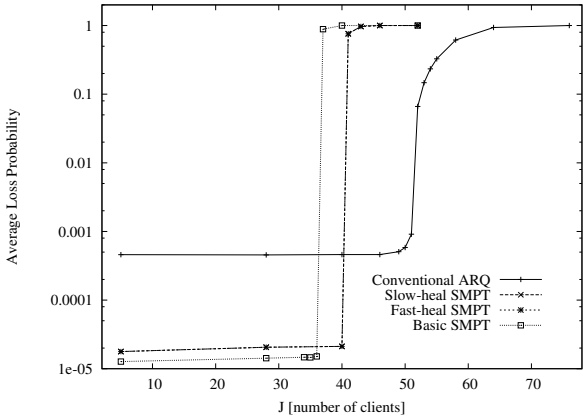

Fig. 96. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ, slow-healing SMPT, fast-healing SMPT, and basic SMPT with $R = 2$ (non-bursty traffic with activity factor $a = 0.8$, the arrivals being Bernoulli process, $B_{\max} = 20$ packets, fixed).

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 SMPT, fast-healing SMPT, and slow-healing SMPT. We have observed that basic SMPT gives marginally smaller loss probabilities than the other considered schemes 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 slow-healing SMPT. Figure 95 compares the performance of different SMPT schemes and the conventional ARQ for a system scenario where the client can use upto $R = 3$ simultaneous codes. The activity factor of each of the clients is 1. Figure 96 compares the different schemes for a system where the clients can use upto $R = 2$ simultaneous codes, and the activity factor of the clients is 0.8. We can observe from the figures that fast-healing SMPT and basic SMPT give marginally smaller loss probabilities in the stable region. However, this comes at an expense of a very slightly reduced capacity (the number of users in the system that can be supported). In Figure 96 it can be seen that the line corresponding to slow-healing overlaps with the line corresponding to fast-healing. This is because of the fact that for $R = 2$, both fast-healing and slow-healing SMPT function in the same manner.

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 quite closely and thus provides a fairly 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

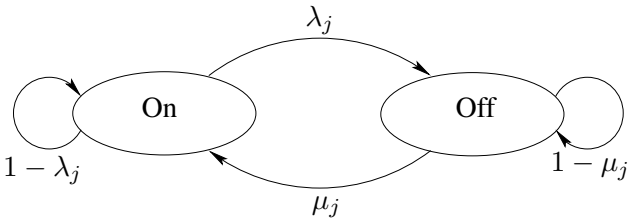

Fig. 97. The bursty traffic at the link layer of each client is modeled with an independent two-state Markov Chain.

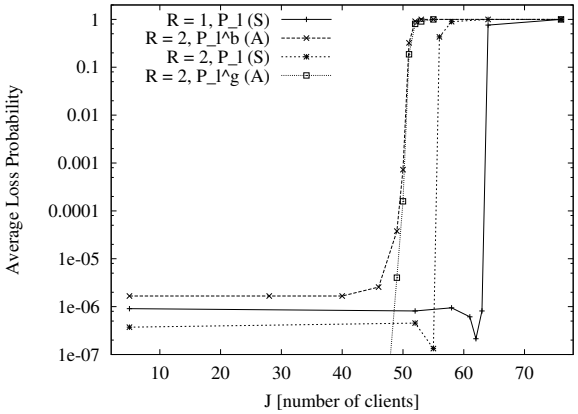

Fig. 98. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (non-bursty traffic with activity factor $a = 0.5$, the arrivals being Bernoulli process, $B_{\max} = 20$ packets, fixed).

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 multi-code 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 non-bursty 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 (i) bursty application traffic [82], [83], and (ii) 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 [18], [19] we model the bursty traffic at the link layer of each client j, $j = 1, \ldots, J$, by an independent two-state Markov chain with the states "on" and "off", as illustrated in Figure 97. These Markov chains make state transitions at the end of every downlink subslot. The state transition probabilities are denoted by λ_j and μ_j as illustrated in Figure 97. 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 generation in a slot is $\mu_j/(\lambda_j + \mu_j)$. Also, note that

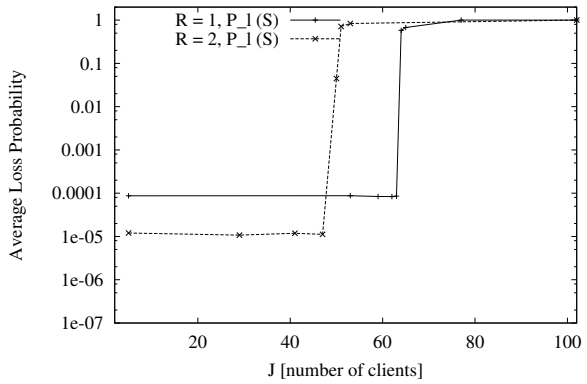

Fig. 99. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda+\mu) = 0.5$ and $1/\lambda = 10$, $B_{\max} = 20$ packets, fixed).

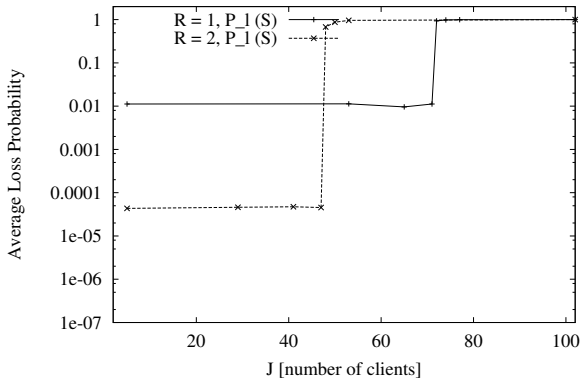

Fig. 100. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda+\mu) = 0.5$ and $1/\lambda = 100$, $B_{\max} = 20$ packets, fixed).

the average burst length of $1/\lambda_j$ packets increases as λ_j decreases. In other words, for small λ_j the traffic becomes more bursty.

In Figures 98, 99, and 100 we compare the average loss probabilities achieved by SMPT and conventional ARQ for non-bursty and bursty traffic. In these cases, the long run average packet generation probability in a slot is set to 0.5, i.e., $a_j = 0.5$ for the non-bursty traffic and $\mu_j/(\lambda_j+\mu_j) = 0.5$ for the bursty traffic for all clients j, $j = 1, \ldots, J$. We observe from Figure 98 that with this low utilization of its single code channel conventional ARQ achieves loss probabilities almost as small as SMPT for non-bursty traffic. For bursty traffic, on the other hand, SMPT achieves significantly smaller loss probabilities, as Figures 99 and 100 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. We also observe similar conditions for clients with different buffer sizes and also the long range packet generation probability in Figures 101 through 106.

Figure 107 compares the performance of the different SMPT schemes in a system where the clients generate bursty traffic. The steady state activity factor ($\mu/(\lambda+\mu)$) is 0.5 and the average burst length is 100 packets. As mentioned earlier, the performance of basic SMPT is marginally better than that of slow-healing SMPT or fast-healing SMPT, obtained at an expense of slightly reduced capacity. Since $R = 2$, both fast-healing SMPT and slow-healing SMPT behave in the same manner and hence both the curves overlap.

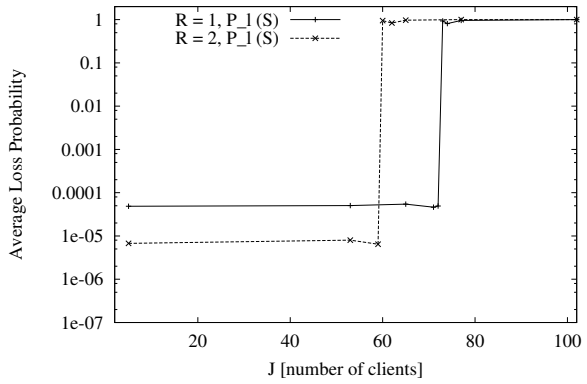

Fig. 101. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.4$ and $1/\lambda = 10$, $B_{\max} = 20$ packets, fixed).

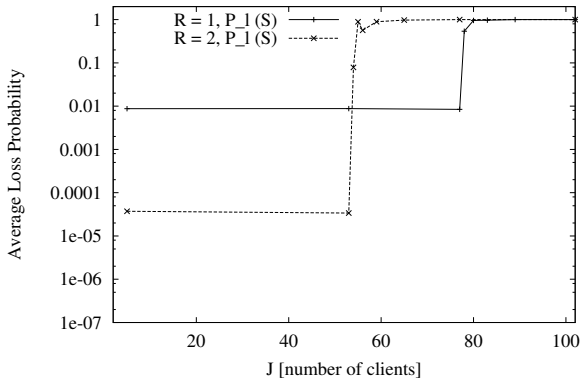

Fig. 102. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.4$ and $1/\lambda = 100$, $B_{\max} = 20$ packets, fixed).

VIII. Conclusions

We have studied Simultaneous MAC Packet Transmission (SMPT), a novel class of ARQ mechanisms for multi-code CDMA systems. We found that SMPT significantly stabilizes the throughput and reduces the 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 flows that are supported in a shared interference environment (cell, cluster). We have developed an analytical framework which accurately characterizes the trade off 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 non-bursty 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, where each client generates traffic that can be carried by a single CDMA code (provided there are no link errors). In ongoing work we are extending the framework to multi-rate traffic scenarios, where 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

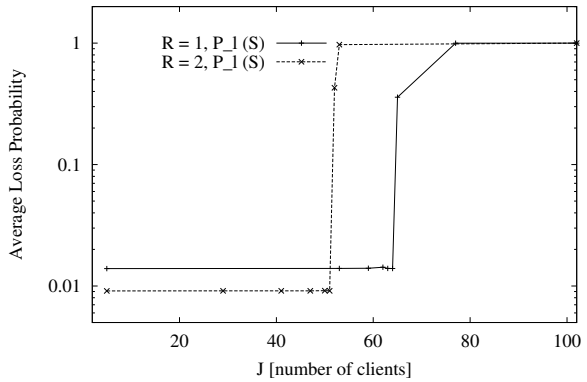

Fig. 103. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.5$ and $1/\lambda = 10$, $B_{\max} = 5$ packets, fixed).

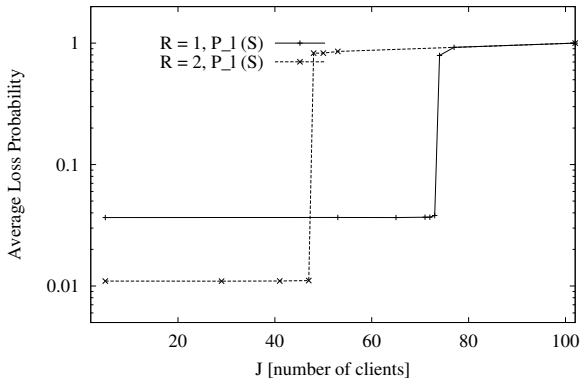

Fig. 104. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.5$ and $1/\lambda = 100$, $B_{\max} = 5$ packets, fixed).

building blocks for the analysis of more general traffic patterns. 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 from 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.

ACKNOWLEDGMENT

We are grateful to Prof. Adam Wolisz for his support of this work and for hosting M. Krishnam at the TKN institute of the Technical University Berlin, Germany, in the summer of 2001.

REFERENCES

[1] EIA/TIA-95 Rev. B, "Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular systems," 1997.

[2] UMTS 30.03, "Universal Mobile Telecommunications System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS," .

[3] F. Fitzek, R. Supatrio, A. Wolisz, M. Krishnam, and M. Reisslein, "Capacity and QoS for streaming video applications over TCP in CDMA based networks," in *Proc. of International Conf. on 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 Communications Magazine*, vol. 9, no. 5, pp. 100–110, Oct. 2002.

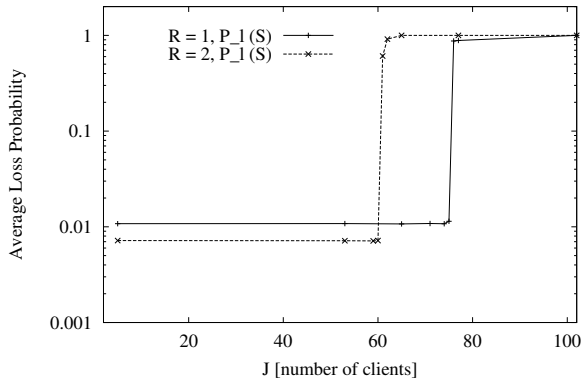

Fig. 105. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.4$ and $1/\lambda = 10$, $B_{\max} = 5$ packets, fixed).

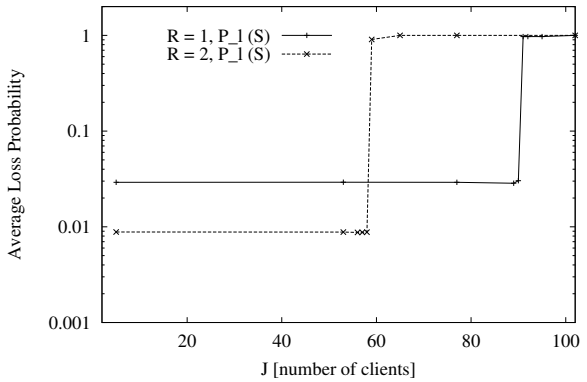

Fig. 106. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ ($R = 1$) and SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.4$ and $1/\lambda = 100$, $B_{\max} = 5$ packets, fixed).

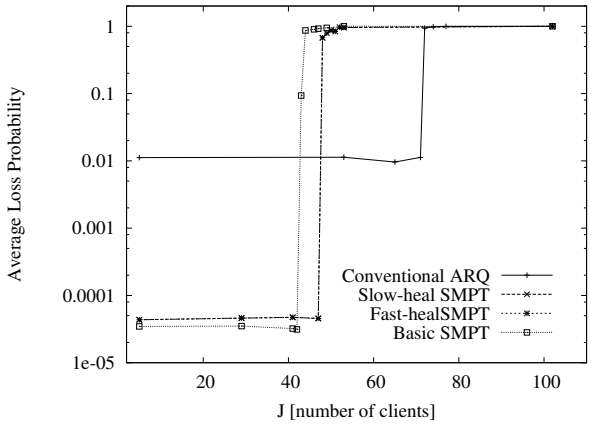

Fig. 107. Average loss probability as a function of number of supported clients (flows) J for conventional ARQ, slow-healing SMPT, fast-healing SMPT, and basic SMPT with $R = 2$ (bursty traffic with $\mu/(\lambda + \mu) = 0.5$ and $1/\lambda = 100$, $B_{\max} = 20$ packets, fixed).

[5] D. Ayyagari and A. Ephremides, "Cellular multicode CDMA capacity for integrated (voice and data) services," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 928–938, May 1999.

[6] G. Brussaard, "Erlang capacity of ATM-based CDMA satellite system," *IEEE Electronics Letters*, vol. 35, no. 8, pp. 613–14, Apr. 1999.

[7] M. Casoni and M. L. Merani, "Erlang Capacity of a TDD-TD/CDMA Architecture Supporting Heterogeneous Traffic," *IEEE Communications Letters*, pp. 468–72, Dec. 2001.

[8] W. Choi and J. Y. Kim, "Forward-link capacity of a DS-CDMA system with mixed multirate sources," *IEEE Trans. on Vehicular Tech.*, vol. 50, no. 3, pp. 737–749, May 2001.

[9] C. S. Kang and D. K. Sung, "Capacities of spectrally overlaid single-code and multicode CDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 839–854, Sept. 2002.

[10] D. K. Kim and F. Adachi, "Theoretical Analysis of Reverse Link Capacity for an SIR-Based Power-Controlled Cellular CDMA System in a Multipath Fading Environment," *IEEE Trans. on Vehicular Tech.*, vol. 50, no. 2, pp. 452–464, Mar. 2001.

[11] D. K. Kim and S.-H. Hwang, "Capacity analysis of an uplink synchronized multicarrier DS-CDMA system," *IEEE Communications Letters*, vol. 6, no. 3, pp. 99–101, Mar. 2002.

[12] J. Y. Kim, G. L. Stuber, and I. F. Akyildiz, "A simple performance/capacity analysis of multiclass macrodiversity CDMA cellular systems," *IEEE Trans. on Comm.*, vol. 50, pp. 304–308, Feb. 2002.

[13] D. K. Kim and D. K. Sung, "Capacity Estimation for a Multicode CDMA System with SIR-based power control," *IEEE Trans. on Vehicular Tech.*, vol. 50, no. 3, pp. 701–710, May 2001.

[14] D. Shen and C. Ji, "Outage evaluation under traffic heterogeneity in CDMA networks," in *IEEE Global Telecommunications Conference*, 2001, vol. 6, pp. 3709–3713.

[15] M. E. Anagnostou and E. N. Protonotarios, "Performance analysis of the selective repeat ARQ protocol," *IEEE Transactions on Communications*, vol. 34, no. 2, pp. 127–135, Feb. 1986.

[16] R. Fantacci, "Queueing analysis of the selective repeat automatic repeat request protocol for wireless packet networks," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 258–264, May 1996.

[17] R. Fantacci, "Mean packet delay analysis for selective repeat automatic repeat request protocol with correlated arrivals and deterministic and nondeterministic acknowledgement delays," *Telecommunications Systems*, vol. 9, pp. 41–57, May 1998.

[18] J. G. Kim and M. M. Krunz, "Bandwidth allocation in wireless networks with guaranteed packet loss performance," *IEEE Trans. Networking*, vol. 8, no. 3, pp. 337–349, June 2000.

[19] J. G. Kim and M. M. Krunz, "Delay analysis of Selective Repeat ARQ for a Markovian source over a wireless channel," *IEEE Trans. on Vehicular Tech.*, vol. 49, no. 5, pp. 1968–81, Sept. 2000.

[20] A. G. Konheim, "A queueing analysis of two ARQ protocols," *IEEE Transactions on Communications*, vol. 28, no. 7, pp. 1004–1014, July 1980.

[21] M. M. Krunz and J. G. Kim, "Fluid analysis of delay and packet discard performance for QoS support in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 2, pp. 384–395, Feb. 2001.

[22] C. H. C. Leung, Y. Kikumoto, and S. A. Sorensen, "The throughput efficiency of the Go-Back-N ARQ scheme under Markov and related error structures," *IEEE Transactions on Communications*, vol. 36, no. 2, pp. 231–234, Feb. 1988.

[23] R. Luo and P. Fan, "Capacity Evaluation of Packet-Switched CDMA System," *IEEE Communications Letters*, pp. 334–336, Aug. 2001.

[24] D.-L. Lu and J.-F. Chang, "Performance of ARQ protocols in nonindependent channel errors," *IEEE Trans. on Comm.*, vol. 41, no. 5, pp. 721–730, May 1993.

[25] Z. Rosberg and N. Shacham, "Resequencing delay and buffer occupancy under the selective-repeat ARQ," *IEEE Trans. Inform. Theory*, vol. 35, pp. 166–173, Jan. 1989.

[26] N. Shacham and D. Towsley, "Resequencing delay and buffer occupancy under selective repeat ARQ with multiple recievers," *IEEE Trans. Comm.*, vol. 39, pp. 928–936, June 1991.

[27] P. M. Soni and A. Chockalingam, "Performance Analysis of UDP with Energy Efficient Link Layer on Markov Fading Channels," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 769–780, Oct. 2002.

[28] W. Turin and M. Zorzi, "Performance Analysis of Delay-Constrained Communications Over Slow Rayleigh Fading Channels," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 801–807, Oct. 2002.

[29] M. Zorzi, "Some results on error control for burst-error channels under delay constraints," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 1, pp. 12–24, Jan. 2001.

[30] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Trans. on Vehicular Tech.*, vol. 46, no. 2, pp. 445–455, May 1997.

[31] M. Elaoud and P. Ramanathan, "Adaptive use of error-correcting codes for real-time communication in wireless networks," in *Proceedings of IEEE Infocom '98*, San Francisco, CA, Apr. 1998, pp. 548–555.

[32] H. Liu, H. Ma, M. ElZarki, and S. Gupta, "Error control schemes for networks: An overview," *Mobile Networks and Applications*, vol. 2, no. 2, pp. 167–182, Oct. 1997.

[33] M. Rice and S. B. Wicker, "Adaptive error control for slowly varying channels," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 917–925, Feb./ Mar./ Apr. 1994.

[34] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabani, and R. D. Gitlin, "AIRMAIL: A link layer protocol for wireless networks," *Wireless Networks*, vol. 1, pp. 47–60, Feb. 1995.

[35] J. B. Cain and D. N. McGregor, "A recommended error control architecture for ATM networks with wireless links," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 16–28, Jan. 1997.

[36] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1971–1988, July 2001.

[37] P. Lettieri, C. Fragouli, and M.B. Srivastava, "Low power error control for wireless links," in *Proceedings of ACM/IEEE MobiCom '97*, 1997, pp. 139–150.

[38] S. Lin and Jr. D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.

[39] H. Minn, M. Zeng, and V. K. Bhargava, "On ARQ scheme with adaptive error control," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 6, pp. 1426–1436, Nov. 2001.

[40] Z. Sun, X. Jia, S. Kimura, and Y. Ebihara, "On-demand QoS hybrid ARQ for multimedia WATM networks," in *Proceedings of IEEE WCNC*, Orlando, FL, Sept. 2002, pp. 625–629.

[41] L. Zhao, J. W. Mark, and Y. C. Yoon, "Highly reliable data transmission using concatenated coding and hybrid-type I ARQ in DS-CDMA systems," in *Proceedings of IEEE Globecom*, San Antonio, TX, Nov. 2001, pp. 3523–3527.

[42] S. Choi and K. G. Shin, "A class of adaptive hybrid ARQ schemes for wireless links," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 3, pp. 777–790, Mar. 2001.

[43] F. Cameron, M. Zukerman, and M. Gitlits, "Adaptive transmission parameters optimisation in wireless multi-access communication," in *Proc. of IEEE International Conference on Networks*, 1999, pp. 91–95.

[44] D. Eckhardt and P. Steenkiste, "Improving wireless LAN performance via adaptive local error control," in *Proc. of Sixth International Conference on Network Protocols*, 1998, pp. 327–338.

[45] S.-H. Hwang, B. Kim, and Y.-S. Kim, "A hybrid ARQ scheme with power ramping," in *Proc. of IEEE Vehicular Technology Conference*, Fall 2001, pp. 1579–1583.

[46] H. Liu and M. El Zarki, "Performance of H.263 video transmission over wireless channels using hybrid ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp. 1775–1786, Dec. 1997.

[47] I. Joe, "An adaptive hybrid ARQ scheme with concatenated FEC codes for wireless ATM," in *Proceedings of ACM/IEEE MobiCom '97*, 1997, pp. 131–138.

[48] 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 *Proceedings of IEEE International Symposium on Wireless Personal Multimedia Communications*, 2002, pp. 668–672.

[49] T. Shoji, O. Kato, and M. Uesugi, "Wireless access method to ensure each user's QoS in unpredictable and various QoS requirements," *Wireless Personal Communications*, vol. 22, pp. 139–151, 2002.

[50] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M.M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 524–534, June 2002.

[51] S. Aramvith, C.-W. Lin, S. Roy, and M.-T. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 558–565, June 2002.

[52] P.-C. Hu, Z.-L. Zhang, and M. Kaveh, "Channel condition ARQ rate control for real-time wireless video under buffer constraints," in *Proceedings of IEEE International Conference on Image Processing*, 2000, pp. 124–127.

[53] 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. of IEEE Vehicular Technology Conference*, Fall 2001, pp. 615–619.

[54] G. Wu, H. Harada, K. Taira, and Y. Hase, "An integrated transmission protocol for broadband mobile multimedia communication systems," in *Proc. of IEEE Vehicular Technology Conference*, 1997, pp. 1346–1350.

[55] M.-S. Do, Y. Park, and J.-Y. Lee, "Channel assignment with QoS guarantees for a multiclass multicode CDMA system," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 934–948, Sept. 2002.

[56] M.R. Hueda, C. Rodriguez, and C. Marques, "Enhanced-performance video transmission in multicode CDMA wireless systems using a feedback error control scheme," in *Proceedings of IEEE Globecom*, San Antonio, TX, Nov. 2001, pp. 619–626.

[57] P.-R. Chang and C.-F. Lin, "Wireless ATM-based multicode CDMA transport architecture for MPEG-2 video transmission," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1807–1824, Oct. 1999.

[58] 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," *Computer Communications*, vol. 21, no. 14, pp. 1256–1268, Sept. 1998.

[59] H. Liu, M.J. Karol, M. ElZarki, and K. Y. Eng, "Channel access and interference issues in multi-code DS-CDMA wireless packet (ATM) networks," *Wireless Networks*, vol. 2, no. 3, pp. 173–192, 1996.

[60] Z. Liu, M. Karol, M. ElZarki, and K. Eng, "Distributed-queuing request update multiple access (DQRUMA)," in *IEEE International Conference on Communications (ICC '96)*, Seattle, WA, June 1995, pp. 1224–1231.

[61] J. Chin-Lin and S. Nanda, "Load and interference based demand assignment (LIDA) for integrated services in CDMA wireless systems," Tech. Rep., Lucent Technologies, 1997.

[62] R.P. Ejzak, D.N. Knisely, S. Kumar, S. Laha, and S. Nanda, "BALI: A solution for high speed CDMA data," Tech. Rep., Bell Labs, 1997.

[63] J. Sadowsky, "Voice and data traffic over the WCDMA air interface," Presentation at ASU, Apr. 2002, `http://www.eas.asu.edu/~trcsemnr/ASU_Talk.ppt`.

[64] F. Wegner, "Personal Communication. Siemens AG, Mobile Radio Access Simulation Group, Berlin, Germany," May 2000.

[65] M. Zorzi, "Data-link packet dropping models for wireless local communications," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 4, pp. 710–719, July 2002.
[66] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253–1266, Sept. 1960.
[67] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Systems Technical Journal*, vol. 42, pp. 1977–1997, Sept. 1963.
[68] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs," *ACM Wireless Networks*, vol. 3, pp. 91–102, 1997.
[69] H. S. Wang and N. Moayeri, "Finite-state markov model – a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 163–177, Feb. 1995.
[70] M. Zorzi and R.R. Rao, "Error-Constrained Error Control for Wireless Channels," *IEEE Personal Communications*, pp. 27–33, Dec. 1997.
[71] 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 *Proceedings of IEEE International Conference on Universal Personal Communications*, Nov. 1995, pp. 211–215.
[72] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmissions over fading channels," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1468–77, Nov. 1998.
[73] R. R. Rao, "Higher Layer Perspectives on Modeling the Wireless Channel," in *Proc. of IEEE Information Theory Workshop*, 1998.
[74] G. L. Stüber, *Principles of Mobile Communications*, Kluwer Academic Publishers, second edition, 2001.
[75] J. Holtzman, "A Simple, Accurate Method to Calculate Spread Spectrum Multiple Access Error Probabilities," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 461–464, Mar. 1992.
[76] L. Kleinrock, *Queueing Systems, vol. 1 Theory*, vol. 1, John Wiley and Sons, 1975.
[77] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.
[78] C. Krasic, K. Li, and J. Walpole, "The case for streaming multimedia with TCP," in *Proceedings of 8th International Workshop on Interactive Distributed Multimedia Systems (IDMS)*, Lancaster, UK, Sept. 2001.
[79] A. Goel, B. Krasic, K. Li, and J. Walpole, "Supporting low-latency TCP-based media streams," in *Proceedings of IEEE International Workshop on Quality of Service (IWQoS)*, Miami Beach, FL, May 2002.
[80] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.
[81] H. M. Chaskar, T. V. Lakshman, and U. Madhow, "TCP over wireless with link level error control: Analysis and design methodology," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 605–615, Oct. 1999.
[82] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.
[83] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 2, pp. 1566–1579, Feb. 1995.

APPENDIX

Appendix A: Evaluation of $\mathcal{N}_c(m,n,b)$ for Fast Healing SMPT

Case I: $2 \le c < R_j$

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

Subcase I.2: $m+b = c-1$ and $n > 2$

Subcase I.3: $\text{mod}(m+b,\ R_j-1) = c-1$ and $n \ge \left\lfloor \frac{m+b}{R_j-1} \right\rfloor + 2$

$$\mathcal{N}_c(m,n,b) = \begin{cases} 2 & \text{if Subcases I.1 and I.3 hold,} \\ 1 & \text{if either of Subcases I.1, I.2, I.3 holds,} \\ 0 & \text{otherwise.} \end{cases}$$

Case II: $c = R_j$

Subcase II.1: $b \ge R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor \ge n \Rightarrow \mathcal{N}_c(m,n,b) = n+1$

Subcase II.2: $b < R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor \ge n \Rightarrow \mathcal{N}_c(m,n,b) = n$

Subcase II.3: $b \ge R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor < n \Rightarrow \mathcal{N}_c(m,n,b) = 1 + \left\lfloor \frac{m+b}{R-1} \right\rfloor$

Subcase II.4: $b < R_j - 1$ and $\left\lfloor \frac{m+b}{R-1} \right\rfloor < n \Rightarrow \mathcal{N}_c(m,n,b) = \left\lfloor \frac{m+b}{R-1} \right\rfloor$

Definitions: $\theta(m,b) = \text{mod}(m+b, R_j - 1)$

$\eta(m,b) = \left\lfloor \frac{m+b}{R_j-1} \right\rfloor$

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

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

Summary:

$$\mathcal{N}_c(m,n,b) = \begin{cases} \delta(b-c+1) + \delta(\theta(m,b)-c+1) \cdot u(n-\eta(m,b)-2) & \text{if } 2 \le c < R_j \\ u(b-R_j+1) + \min(\eta(m,b),\ n-1) & \text{otherwise.} \end{cases}$$

Appendix B: Evaluation of $\mathcal{N}_c(m,n,b)$ for Slow Healing SMPT

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

$$\mathcal{N}_c(m,n,b) = \begin{cases} 3 & \text{if Subcases I.1, I.2 and I.3 hold,} \\ 2 & \text{if two of the subcases I.1, I.2, and I.3 hold,} \\ 1 & \text{if one of the subcases I.2, I.2, and I.3 holds.} \end{cases}$$

Case II: $c = R_j$ and $m + b \le R_j(R_j - 1)/2$
Subcase II.1: $b \ge R_j - 1$ and $m + b < R \cdot (R-1)/2 \Rightarrow \mathcal{N}_c(m,n,b) = 1$
Subcase II.2: $b \ge R_j - 1,\ m + b = R \cdot (R-1)/2$ and $n \ge R \Rightarrow \mathcal{N}_c(m,n,b) = 2$
Subcase II.3: $b < R_j - 1,\ m + b = R \cdot (R-1)/2$ and $n \ge R \Rightarrow \mathcal{N}_c(m,n,b) = 1$
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: $2 \le c < R_j$ and $m + b > R_j(R_j - 1)/2$
Subcase III.1: $b = c - 1$
Subcase III.2: $n \ge c$
Subcase III.3: $\theta_s(m,b)$ and $n \ge \eta_s(m,b)$

$$\mathcal{N}_c(m,n,b) = \begin{cases} 3 & \text{if Subcases III.1, III.2, and III.3 hold,} \\ 2 & \text{if two of the subcases III.1, III.2, and III.3 hold,} \\ 1 & \text{if one of the subcases III.2, III.2, and III.3 holds.} \end{cases}$$

Case IV: $c = R_j$ and $m + b > R_j(R_j - 1)/2$
Subcase IV.1: $b \ge R_j - 1$ and $n \ge R + \eta_s(m,b) \Rightarrow \mathcal{N}_c(m,n,b) = \eta_s(m,b) + 2$
Subcase IV.2: $b \ge R_j - 1$ and $R \le n < R + \eta_s(m,b) \Rightarrow \mathcal{N}_c(m,n,b) = n - R + 2$
Subcase IV.3: $b \ge R_j - 1$ and $n < R \Rightarrow \mathcal{N}_c(m,n,b) = 1$
Subcase IV.4: $b < R_j - 1$ and $n \ge R + \eta_s(m,b) \Rightarrow \mathcal{N}_c(m,n,b) = \eta_s(m,b) + 1$
Subcase IV.5: $b < R_j - 1$ and $R \le n < R + \eta_s(m,b) \Rightarrow \mathcal{N}_c(m,n,b) = n - R + 1$
Subcase IV.6: $b < R_j - 1$ and $n < R \Rightarrow \mathcal{N}_c(m,n,b) = 0$

Definitions:
$$\eta_r(m,b) = \left\lfloor \frac{1+\sqrt{1+8(b+m)}}{2} \right\rfloor$$
$$\theta_r(m,b) = b + m - \eta_r \cdot (\eta_r - 1)/2$$
$$\eta_s(m,b) = \left\lfloor \frac{b+m-R\cdot(R-1)/2}{R-1} \right\rfloor$$
$$\theta_s(m,b) = \text{mod}(b + m - R \cdot (R-1)/2, R-1)$$
$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise.} \end{cases}$$
$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

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

Appendix C: 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 one if a packet is generated in the first slot, and zero otherwise. The backlog at the end of the run of good slots is now given by

$$b_n = \begin{cases} \max\{\max(b_{o,n}+1, B_{\max,j}) + l - g(n),\ 0\} & \text{if } a_s = 1 \\ \max\{b_{o,n} + l - g(n), 0\} & \text{otherwise.} \end{cases} \tag{22}$$

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

$$Pr\{b_n,k,l,m,n,a_s|b_o\} = \begin{cases} \binom{m}{k} a_j^k (1-a_j)^{m-k} \cdot a_j \cdot \binom{n-1}{l} a_j^l (1-a_j)^{n-1-l} \cdot \pi_j(m,n) \\ \qquad \text{if } b_n = \max\{0,\ \max(b_o+k+1,\ B_{\max}) + l - 1 - g(n)\} \text{ and } a_s = 1 \\ \binom{m}{k} a_j^k (1-a_j)^{m-k} \cdot (1-a_j) \cdot \binom{n-1}{l} a_j^l (1-a_j)^{n-1-l} \cdot \pi_j(m,n) \\ \qquad \text{if } b_n = \max\{0,\ \max(b_o+k,\ B_{\max}) + l - 1 - g(n)\} \text{ and } a_s = 0 \\ 0 \text{ otherwise.} \end{cases} \tag{23}$$

Finally, we obtain

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