TCP-Aware Channel Allocation in CDMA Networks

Majid Ghaderi, Ashwin Sridharan, *Member, IEEE,* Hui Zang, *Senior Member, IEEE,* Don Towsley, *Fellow, IEEE,* and Rene Cruz, *Fellow, IEEE*

Abstract— This paper explores the use of rate adaptation in cellular networks to maximize throughput of long-lived TCP sessions. Modern cellular networks incorporate RF technology that allows them to dynamically vary the wireless channel rate in response to user demand and channel conditions. However, the set of data rates as well as the scheduler's rate adaptation policy are typically chosen to maximize the throughput of inelastic connections. We focus on the problem of maximizing the throughput of TCP connections and propose a joint optimization of MAC and physical layer parameters with respect to TCP sending rate. In particular, we propose a simple *TCP-aware channel scheduler* that adapts the wireless channel rate to changes in the TCP sending rate and explore its performance for both single and multiple concurrent sessions. In the case of a single TCP session, we develop a fluid model of its steady-state behavior in such a system that adapts between *two* channel rates. The accuracy of the model, its utility in selecting *optimal rates* as well as the performance of systems with up to three channel rates are explored with *ns-2* simulations. Our results indicate that a two-rate scheme improves TCP throughput by 15% to 20% over a system that does not exploit rate adaptation and that little additional benefit accrues from the addition of a third channel rate. Finally, we extend the framework to scenarios where bandwidth is shared by multiple TCP sessions. We propose two channel allocation algorithms, one relying on detailed TCP state information, the other not, and explore their performance through simulation. Our results indicate that TCP throughput is relatively *insensitive* to either channel allocation algorithm and adaptive rate variation is the dominant factor in performance.

Index Terms— Wireless networks, TCP throughput, Rate adaptation, Cross-layer optimization, Scheduling.

I. INTRODUCTION

MODERN digital communication technologies combined with powerful mobile processors now allow wireless channel schedulers in cellular networks to rapidly change the allocated channel resources in response to channel conditions as well as user demands. This is achieved by changing parameters (and combinations thereof) such as the coding rate, spreading factor, modulation scheme and link layer retransmission rate. For ease of exposition, we shall refer to these jointly as *RF control variables*. These variables essentially trade-off data rates for improved *frame error rates* (FER) and vice versa.

Cellular networks typically specify various combinations of the RF control variables that result in a set of allowed data rates and corresponding FER. The RF scheduler dynamically assigns rates from this allowed set based on its rate adaptation policy. For example, in the CDMA2000 1xRTT network [1], the scheduler can dynamically transition between five different data rates during a mobile's session in response to buffer

content and channel conditions by varying the spreading factor through the *Walsh code*[1] length. A shorter Walsh code lowers the spreading factor which results in higher data rates. However, a shorter code results in lower SINR (Signal to Interference and Noise Ratio), and supports fewer users simultaneously.

In practice, the above mentioned parameters as well as the scheduler's rate adaptation policy are chosen to optimize the *raw* physical layer goodput of a user. The set of data rates is obtained by choosing a combination of the RF control variables that produces the highest channel data rate under a particular channel condition[2] for a target frame error rate. Similarly, scheduling policies typically assign the highest possible data rate allowed (for a given channel condition) from the set of given data rates that can clear the buffer backlog[3](see [2] for a detailed characterization of this behavior). While ideal for *inelastic* constant rate applications, this resource allocation methodology can produce sub-optimal performance of *elastic* applications and protocols, in particular TCP, that adapt their rate in response to feedback from the receiver.

As is well known, TCP, by far the most dominant transport protocol, uses an additive increase multiplicative decrease (AIMD) algorithm that gradually increases its transmission rate based on receiver feedback and rapidly throttles back when it perceives losses (either due to congestion or channel errors). Given this complex relation between TCP throughput and the channel transmission and loss rate, the same trade-off in channel capacity and frame error rate that works for inelastic traffic may yield data rates and FERs that degrade TCP throughput. Similarly, a scheduler's rate adaptation policy that always aims to clear buffer backlog can be sub-optimal for TCP. For example, when the TCP source has a small window and is ramping up its rate, it is very sensitive to losses but not to the assigned channel rate. In such a state, if the RF scheduler allocates a high channel rate at the expense of a high FER (perhaps due to a sudden accumulation of buffer backlog as a result of jitter in the network), the TCP source cannot fully utilize the high rate and in fact may drop its window or time out due to the channel errors. Conversely, for large windows (high TCP sending rates), it may be advisable to allocate high channel rates even at the expense of high bit error rates, since a low channel rate will inevitably result in packet loss due to congestion. On a related note, a previous study found that sharp bandwidth oscillations induced by rate

---

[1]In CDMA systems, the Walsh code is an orthogonal code used to identify each user and mitigate interference from other users.

[2]A channel condition is defined by a particular range of SINR values and is a function of fading, interference, *etc.*

[3]The CDMA2000 1xRTT is an example of a commercial system with such features.

adaptation of the RF scheduler in CDMA networks that are agnostic to TCP result in throughput degradation [3].

Finally, as noted previously, in current CDMA networks (*e.g.*, CDMA2000 1xRTT) the high-rate channels are achieved by reducing the *spreading factor* of the orthogonal Walsh codes. This essentially implies that only a *few* users can simultaneously share the high rate channels at any given time, simply because of the shorter code-length. In particular, the *higher* the channel rate, the *fewer* the number of concurrent users that can be supported. Thus selection as well as allocation of high rate channels must be made in a judicious fashion that not only accounts for TCP behavior but is also fair across users.

The above discussion clearly motivates the case for *TCP-aware* dynamic rate allocation as a means to increase throughput of TCP sessions over wireless channels[4]. In order to achieve this objective, such a scheduler should be able to: a) choose control variables such as coding rate *etc.* that yields the optimal set of data rates (and corresponding frame error rates) from the perspective of TCP throughput, and b) set channel rates in a manner that is cognizant of TCP dynamics. The proposal of a simple scheduler that captures the above mentioned properties and its analysis form the main objective of this paper.

Specifically, we propose a wireless channel scheduler that allocates different channel rates from a set of optimized rates to TCP sessions in response to their sending rates. In the context of a single TCP session, we develop a model to compute its long-term throughput under such a scheduler and use it for joint optimization of control variables, *e.g.*, spreading factor, to compute the set of optimal channel rates that are used by the scheduler. We extend the rate allocation framework to incorporate the presence of multiple TCP sessions. In particular, given a set of channel rates that are computed by the single flow model, we study different mechanisms for allocating these channel rates to different TCP flows in order to improve the throughput capacity of the system.

Our contributions can be summarized as follows:

1) We explore the cross-layer optimization of MAC and physical layer parameters with respect to TCP sending rate. At the MAC layer, we propose a scheduler that determines the rate to allocate to a TCP session from a set of channel rates. At the physical layer, we explore the set of channel rates that maximizes TCP throughput.

2) For a single TCP session, we develop an analytical expression for the steady-state throughput of a long-lived TCP session in such an environment. Our model explicitly captures the dependency of the scheduler on TCP sending rate as well as the impact of the presence of two distinct rates and frame error rates on TCP. Each channel transmission rate results in a different round trip time and FER. This system is used to study the benefits of cross-layer optimization of the dynamic rate adaptation feature of modern cellular networks with respect to TCP throughput. Indeed, a key benefit of an analytical model is the significant speed-up in the numerical optimization process compared to lengthy simulations. In some instances studied in this paper, using *ns-2* simulations to find the optimal configuration required several days of computations, while the same optimization took just a few minutes using our model.

3) We demonstrate how the analytical expression can be used to choose RF control variables that maximize TCP throughput. For example, we identify the optimal coding rates to be used in each of the two states when coding rate is used to control data rate and FER. The model is also applied to determine the optimal spreading factors, which is representative of rate control in current CDMA networks. Our studies show that throughput improvements of the order of 15% to 20% can be obtained for a single TCP session. Furthermore, most of benefits of adaptive rate allocation are obtained with just two channel rates.

4) Finally, we study the benefits of dynamic rate adaptation for multiple TCP sessions in a CDMA system, in particular the presence of statistical multiplexing gain. We propose two multi-user channel allocation mechanisms for such an environment: one, which relies only on coarse TCP dynamics, called $\alpha-$EMPT scheduler, and another which takes more detailed TCP rate information into consideration, called $\beta-$AWARE. The behavior of the system under these two schedulers is explored via extensive simulations over the space of input parameters α and β (to be explained later). Our results indicate that multi-user channel allocation strategies have only a limited impact on TCP, and most benefits come from dynamic rate adaptation.

The remainder of the paper is structured as follows. Section II discusses the related work in more depth and contrasts it with our current work. In Section III, we describe our system model and state our assumptions regarding different aspects of the system operation. The proposed channel schedulers for single-user and multi-user systems are presented in Section IV. Section V presents our model for a TCP session that captures the correlation between the TCP and the scheduler. Section VI presents our numerical results. We validate the accuracy of the model against *ns-2* simulations and also explore the impact of rate adaptation on single and multi-TCP sessions through the model and simulations. Our conclusions as well as future work are discussed in Section VII.

II. RELATED WORK

Numerous approaches have been proposed in the literature to optimize TCP performance in wireless networks. These approaches can be broadly categorized as either *TCP enhancement* approaches or *link-layer* optimization approaches.

The framework presented in this paper is a link-layer optimization approach that, rather than modify TCP to adapt to RF dynamics (as with TCP enhancement mechanisms), adapts the RF layer to TCP dynamics. In this view, our work is closer in philosophy to previous literature that optimizes link layer parameters like Forward Error Correction (FEC), Automatic Repeat reQuest (ARQ) and RF scheduling to improve TCP throughput.

---

[4]Such an approach falls in the realm of RF cross-layer optimization.

References [4] and [5] analyzed the trade-off between TCP throughput and the amount of FEC added by the link layer. They showed that there exists a coding rate that maximizes TCP throughput though they only considered channel error losses. Reference [6] also conducted a similar study but included the impact of signal power and ARQ as well. Baccelli *et al.* [7] developed an analytical model of TCP that includes the impact of congestion losses due to a finite capacity channel. They used this model to study the impact of both coding rate and processing gain on TCP throughput. See also [8]–[10] for other cross-layer optimization techniques proposed to improve TCP throughput. All these previous studies however consider a static scenario with only a *single* coding rate.

Adaptive coding on the fly has been studied in [5] and [11] via simulations. In both cases, however, the scheduler behavior is agnostic to TCP instantaneous state and is based on expressions for the long term TCP throughput.

References [12] and [13] modeled TCP in the presence of variable round trip times and packet losses on wireless links. However, they assume that the variability is independent of TCP dynamics which is at odds with the environment considered here as well as the commercial environment studied in [2]. Finally, the authors of [14] studied optimization of transmission power to maximize TCP throughput. They explicitly considered TCP dynamics in the selection of the transmission power level. However the resulting solutions were quite complex, requiring detailed TCP state knowledge. Furthermore, our focus is also different from this work since we study the impact of rate adaptation for both single and multiple TCP sessions which was not considered in [14].

III. System Model

In this section, we motivate the problem and present the system model including assumptions regarding different aspects of the system operation. For ease of exposition, we shall use the CDMA2000 1xRTT [1] system as an example of a practical system suited to our proposed adaptive resource allocation mechanism, although our schemes also apply equally to other wireless systems that dynamically adapt wireless channel rates in response to user sending rates.

Fig. 1 depicts the wireless hop in a typical cellular network. It comprises a base station, mobile devices, and a buffer at the base station for each user. The channel scheduler resides at the base station (or in the case of CDMA2000 1xRTT, the Base Station Controller) and determines the rate allocated to each mobile session. For the purposes of this work, we focus on mobile sessions that involve TCP bulk transfers on the downlink.

The channel scheduler is assumed to have the ability to dynamically assign different wireless transmission rates (termed *channels*) to each user in a sector. For example, CDMA2000 1xRTT supports five different rates ranging from 9.6 Kbps to 153.6 Kbps per user by changing processing gain and power to trade-off channel error for increased capacity. The main theme of this work is to show that TCP throughput can be increased by exploiting this feature, in particular through
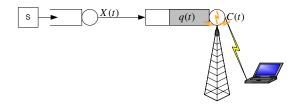

Fig. 1. Illustration of a cellular hop

careful selection of the channel rates and a mechanism to adaptively allocate different channel rates in response to TCP state.

Rate adaptation is also beneficial when *multiple* TCP sessions are in progress. Through a detailed analysis of a single TCP session in a multi-rate scenario, we show that the higher rate channel is not *always* utilized. This immediately suggests that by virtue of statistical multiplexing gain, with a few high rate channels and careful allocation of these channels in a multi-session system, TCP sessions can achieve nearly similar throughput as isolated sessions with dedicated channels. We verify discuss this in Section VI where it is shown that this framework and our proposed channel allocation schemes yield bandwidth/enery savings and higher throughput compared to a network with a single static channel rate.

Such a framework is also practically relevant. Again taking CDMA2000 as an example, high rate channels are achieved by reducing processing gain (equivalently shortening the Walsh orthogonal code). This makes them much more susceptible to interference compared to low-rate channels, and consequently, only a small number of users may use a high rate channel simultaneously. For example, 30 users may simultaneously use a 9.6 Kbps channel but only two users may concurrently be assigned a 102.4 Kbps channel. This is very similar to our framework wherein several TCP sessions contend for a few high rate channels and hence our results are applicable to such a system.

Next we present the system model and assumptions utilized in our work that capture the various aspects of a CDMA system mentioned above:

1) For simplicity, we focus on a *two-rate* system to explain our rate allocation framework. Extensions of the system model to three or more rates is straightforward. Denote by C_0 and C_1 the low-rate (also called *fundamental*) channel rate and high-rate (also called *supplemental*) channel rates respectively. Clearly $C_0 \leq C_1$. At each point in time, the scheduler decides which of the two channels, the low rate channel (rate C_0) or the high rate channel (rate C_1), are to be assigned to a TCP session. Each active user can always be assigned a low-rate channel, however allocation of high-rate channels is arbitrated by the channel scheduler because there are only a few high rate channels.

2) The packet error probability is implicitly assumed to be a function of the assigned rate and denoted by $p_0(p_1)$ when the assigned channel rate is $C_0(C_1)$. This is an important feature representative of current wireless systems where an increase in channel rate typically

comes at the cost of increased packet error probability. For simplicity, we shall refer to (p_i, C_i) together as a *state* or *mode*. We shall dwell on the relation between p and C in subsection VI-A.

3) We assume the presence of power control to primarily combat fast fading and interference effects. This is true in current systems where fast closed-loop power control tracks a specified target SINR (or equivalently target FER).

4) We assume no (or a very small) buffer at the base station. Hence, TCP experiences congestion if its sending rate exceeds the maximum assigned channel rate.

5) Finally, in the context of multiple TCP sessions, we assume a fixed population of N sessions. Each session can be assigned a low-rate channel, however, only $K \leq N$ sessions can be assigned a high-rate channel concurrently. This captures issues such as increased interference in high-rate channels and also allows us to study statistical multiplexing gain by varying K.

Finally, we emphasize that at this stage, no specific assumptions have been made regarding how the channel rates are achieved nor how they result in the specific channel error probabilities. Indeed the specific relation is not required in our TCP model and only the actual variables (p_i, C_i) are required. The channel rates and packet error probabilities are a function of the underlying technology that is used, *e.g.*, adaptive modulation and spreading. This issue is addressed in detail in Section VI.

IV. TCP-AWARE RESOURCE ALLOCATION

In this section we present our channel schedulers for single-user and multi-user systems. We first describe the single-user scheduler and based on which we then propose two multi-user schedulers.

A. Single-User Scheduler

The TCP-aware channel scheduler we consider resides at the base station. It is quite similar in operation to the 1xRTT scheduler described in Section III, with the exception that it assigns a channel rate based on the *user's TCP sending rate*[5] rather than its buffer content. Specifically, whenever the TCP sending rate exceeds (or drops below) the current channel rate, the base station increases (decreases) the channel rate (accompanied by the corresponding FER, signal power, *etc.*). If the session sending rate exceeds the maximum possible channel rate, the session experiences packet loss due to dropped packets.

As before, we utilize a *two-rate* system as the reference example. We note that modern cellular systems have four or more modes, *i.e.*, they can support up to four or more different channel rates. However, obtaining succinct analytical expressions for TCP throughput even for three rates is quite difficult. Moreover, our simulations show that there is only a small improvement in system performance by considering

---

[5]This of course implies the assumption that the scheduler can measure TCP sending rate, which is defined in the next section.

more than two rates. We will address this issue in greater detail in Section VI.

The single-user scheduler operates as follows. The scheduler decides which of the two channel rates C_0 and C_1 are to be assigned based upon the user's TCP sending rate. If the TCP sending rate is below C_0, the scheduler assigns a channel rate of C_0, otherwise it assigns C_1. In other words:

$$C(t) = \left\{ \begin{array}{ll} C_0 & \text{if } X(t) \leq C_0 \\ C_1 & \text{otherwise} \end{array} \right.$$

where $X(t)$ is the TCP sending rate at time t and $C(t)$ is the assigned channel rate.

The motivation for such a scheduler was presented in Section I. Intuitively, when the TCP sending rate is small, a lower frame error rate is essential (to avoid time-outs *etc.*). The scheduler can exploit the knowledge of the low TCP sending rate to trade-off channel rate for channel integrity. At higher TCP sending rates, it is more appropriate to assign a larger channel rate at the expense of a higher FER. This is because, even though packet loss probability due to channel errors increases, a larger channel rate prevents packet loss due to congestion, which would have happened with probability *one* were the rate not increased, allowing TCP to transmit at high rates for a longer time.

We note that there are several other features of TCP, for example, timeout values, window size, sending state, *etc.* which could potentially be utilized to improve upon our proposed scheduler. Incorporation of such features however, would make the scheduler complex to implement as well as to study. Our aim here is to propose a system that involves minimal modifications to schedulers used in current technologies like CDMA2000 1xRTT and EV-DO, and can be studied analytically. From this perspective, we believe that our proposal to incorporate knowledge of only TCP sending rate satisfies both goals.

B. Multi-User Scheduler

In this subsection, we propose two flow-level channel schedulers, which are extensions of the single-user scheduler described in the previous subsection, for channel allocation among multiple TCP sessions. We assume that all users have the same propagation delay. This is not unreasonable since all major cellular networks deploy split-TCP [15] on the downlink implying that TCP sessions on the downlink originate at a common proxy, and hence share a common path.

A naive extension of our single-user scheduler would be to have a dedicated high-rate channel for every user in the system. However, our analytical and simulation results for single-user system show that the high-rate channel is not fully utilized implying that such an extension would be wasteful of system resources, *i.e.*, high-rate channels. Hence, we assume that there are only a few high-rate channels as is the case in CDMA2000 systems and propose flow-level scheduling strategies to maximize TCP throughput in a multi-user scenario.

1) $\alpha-$EMPT Scheduler: The $\alpha-$EMPT, which has a control parameter α works as follows:

1) Upon commencement, a TCP session i is assigned a fundamental channel with rate C_0. Note that all N sessions can be assigned fundamental channels.

2) Similar to the single-session case, the scheduler must make a decision on allocation of a high rate channel whenever a TCP session's sending rate reaches C_0.

3) In such a scenario, if a high rate channel is free then it is allocated to the requesting session. Otherwise, one of the K users occupying the high rate channels is *preempted* with probability α and the released channel is allocated to the requesting TCP session. Note that in this case, the preempted session faces congestion loss with probability *one* and halves its sending rate. Alternatively, with probability $(1-\alpha)$ the requesting session is *denied* the high rate channel, in which case it is the requesting session that experiences congestion and drops its sending rate by half.

4) Whenever a user in a high-rate channel drops its sending rate (triggered by either a congestion loss or a random channel error), the user releases the high-rate channel and gets assigned a fundamental channel.

We need to specify the preemption probability α in order to completely specify the scheduler. Furthermore, it is clear that α, the number of supplementary channels as well as the channel rates will all impact TCP throughput. In subsection VI-E, we study these various issues.

2) $\beta-$AWARE Scheduler: The $\beta-$AWARE scheduler is similar to the $\alpha-$EMPT scheduler, with the difference that instead of randomly selecting a user occupying the supplemental channel for preemption, it chooses a user with the largest window size among those whose rate is above a factor β of the peak capacity C_1.

The intuition behind the $\beta-$AWAREscheduler can be explained as follows. Let us assume that user j has a high-rate channel and $X_j(t)$ is the sending rate of user j at time t $(C_0 < X_j(t) < C_1)$. We also assume that at time t, user i requests a high-rate channel, *i.e.*, $X_i(t) = C_0$. Consider the following scenarios:

1) If $X_j(t)$ is close to C_0 then there is no throughput gain in preempting user j to allocate the high-rate channel to user i. In fact, the overall system throughput may even degrade if user j is preempted.

2) If $X_j(t)$ is very close to C_1 then after a short period of time, user j will reach C_1 and faces congestion with probability one. When that happens, the high-rate channel will stay idle until another user makes a request. In this case, system throughput can be improved by preempting user j to accommodate user i at high rate.

$\beta-$AWARE tries to distinguish between these scenarios by predicting how much longer a high-rate user can send at high rate before reaching C_1. $\beta-$AWARE uses a threshold β $(C_0/C_1 \leq \beta \leq 1)$ to decide when to preempt a high-rate user j by considering how close $X_j(t)$ is to C_1. The scheduler operates similar to $\alpha-$EMPT except when contention occurs. Assume user i requests a high-rate channel when all high-rate channels are occupied. Let j^* denote the user that has the highest sending rate, *i.e.*, $j^* = \arg\max_{1 \leq j \leq N} X_j(t)$. Then,

$\beta-$AWARE works as follows:

1) if $X_{j^*}(t) \leq \beta C_1$, then user i's request for a high-rate channel is denied,

2) otherwise, user j^* is preempted and the released high-rate channel is assigned to user i.

Therefore, we have a multi-user scheduler that takes into consideration the sending rate of both low-rate users and high-rate users in making scheduling decisions. Furthermore, note that $\beta = 1$ corresponds to $\alpha = 0$ in the $\alpha-$EMPT scheduler since in both cases this would result in no preemption. Similarly $\beta \leq C_0/C_1$ corresponds to $\alpha = 1$. In subsection VI-E, by varying the parameters α and β, we study a wide range of behaviors when either ignoring or incorporating TCP rate information into channel allocation decisions.

It is worth mentioning that since we consider long-lasting TCP flows in a homogeneous environment where all sessions experience the same propagation delay, the above schedulers are inherently fair. All sessions are statistically treated the same by the schedulers, and hence they achieve the same steady-state throughput.

V. SINGLE TCP SESSION ANALYSIS

In the previous section, we presented a TCP-aware channel scheduler for single and multiple sessions. Next, we present an analytical model for the single session case that explicitly captures the inter-play between TCP and the scheduler in a two-rate system. Such a model can be useful to quickly explore system behavior instead of utilizing lengthy simulations.

From a TCP flow perspective, such a system results in two distinct operating regimes with different channel capacities, round trip times and packet error probabilities. In this section, we present a TCP model for such a system in the single-session case that enables us to study the impact of dynamic channel allocation on TCP throughput. Extending the model to multi-session case is an ongoing work. However, in Section VI, we study the scheduler performance in multi-user case through *ns-2* simulations.

A. TCP Model

Existing TCP models typically assume that the Round Trip Time (RTT) and packet loss statistics are independent of TCP dynamics in throughput calculations. However, in the wireless environment considered in this work, there exists a strong correlation between the scheduler and TCP sending rate. In particular, the channel capacity C that affects RTT, as well as packet loss probability p are functions of the TCP sending rate. As an illustration, Fig. 2 depicts the evolution of window size of TCP in steady-state when serviced by the proposed TCP-aware channel scheduler. The scheduler assigns rates based on the TCP sending rate and as can be seen, this in turn affects the window *growth* rates. In Section VI, we show that ignoring this dependency can result in large errors in throughput prediction.

In order to tackle the impact of the proposed channel scheduler on TCP throughput, we use the model developed by Baccelli *et al.* [7] for a *single fixed rate* as a starting point and develop a model that accounts for the two rate regime. We present a more detailed explanation of our model below:
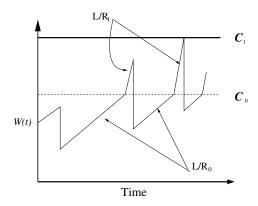
Fig. 2. TCP window size evolution over a variable rate channel.

1) We assume that the TCP version is TCP Reno and model the TCP window growth in steady state as a fluid process where the window size grows linearly in the absence of loss.
2) The sender is assumed to always have data to send and, for analytical tractability, we ignore time-outs and slow start.
3) Let $W(t)$ denote the window size of TCP at time t and $R(t)$ the round trip time at time t. In the absence of a buffer, if the scheduler is in mode $i = 0, 1$ at time t, we approximate the round trip time $R(t)$ as:

$$R(t) = R_i = a + L/C_i$$

where a is the round-trip propagation delay, L is the packet length and C_i is the channel capacity in mode i.
4) Let $X(t)$ denote the instantaneous TCP sending rate at time t in bits/sec. Then $X(t) = \frac{W(t)}{R(t)}$.
5) During congestion avoidance, the TCP window size increases by roughly one packet (L bits) every R_i seconds in mode i when there is no packet loss. We approximate this in our fluid model with a linear growth rate of L/R_i. Consequently, the sending rate grows at a linear rate of L/R_i^2 bits/sec^2 in the absence of loss. To see this note that the rate of increase of $X(t)$ is given by:

$$\frac{X(t + R_i) - X(t)}{R_i} = \frac{W(t + R_i) - W(t)}{R_i^2} = \frac{L}{R_i^2}.$$

6) As in [7], [16], we assume that the channel losses can be modeled by an inhomogeneous Poisson process with rate $p_i X(t)$, $i = 0, 1$ at time t.
7) If the TCP sender is not constrained by the receiver window, then TCP experiences congestion with probability 1 when its sending rate, $X(t)$, exceeds C_1. However, if TCP is constrained by the receiver window size, the sending rate stops increasing once it reaches the receiver advertised window and it experiences only channel related losses.

We need to approach the problem differently depending on whether the two modes (channel rates) satisfy either $C_1 \leq 2C_0$ or $C_1 > 2C_0$. For the important case[6] of $C_1 \leq 2C_0$

[6]In practice, most wireless rates obey the relationship $C_1 = 2C_0$.

we derive expressions for mean TCP throughput for both, the unconstrained rate case as well as the constrained rate case. However, we present results only for the unconstrained rate case in Sections V-B. Interested readers are referred to [17] for the analysis of the rate constrained case. We have also obtained expressions for the TCP throughput for the case when $C_1 > 2C_0$ but omit them here due to lack of space as well as the fact that most performance improvements are obtained in the region $C_1 \leq 2C_0$.

For ease of exposition, we define some notation.

1) Let $f_i(x, t)$ denote the density function of rate $X(t)$ at time t in mode i. Then,

$$f_i(x, t)\Delta x = \mathbb{P}\left\{x \leq X(t) \leq x + \Delta x, C(t) = C_i\right\},$$

where $C(t)$ is the instantaneous channel rate at time t. It is clear from the above definition that

$$f(x, t) = f_0(x, t) + f_1(x, t).$$

where $f(x, t)$ denotes the density function of $X(t)$.
2) We also define the terms:

$$\delta_i = \frac{L}{R_i^2}, \quad \gamma_i = \frac{p_i}{L}, \quad \text{for } i = 0, 1.$$

B. Throughput Analysis

Before proceeding with the analysis, it is worthwhile discussing an important aspect of the variable rate environment that directly affects the analysis, namely the behavior of the TCP sending rate at the channel transition points. Suppose at time t^-, $X(t^-) = C_0$. The corresponding window size is given by $W(t^-) = C_0 R_0$. As per the proposed policy, the scheduler would then assign a rate of C_1 to the TCP session at time t^+ resulting in a new RTT of R_1. Since TCP is a window-based protocol, the window size will be *continuous* at the rate transition point. Specifically, we have $W(t^+) = W(t^-) = C_0 R_0$. Consequently, the new sending rate is

$$X(t^+) = \frac{W(t^+)}{R_1} = \frac{R_0 \cdot C_0}{R_1} = \frac{R_0}{R_1} X(t^-).$$

In other words, the TCP sending rate experiences a discontinuous jump by a factor of $g = R_0/R_1$ when the channel capacity transitions from C_0 to C_1. Similar arguments can be used to show that if $X(t) \geq C_0$ and TCP experiences a loss, the sending rate drops by a factor $1/2g$. This aspect of the TCP sending rate must be accounted for in the analysis.

For purposes of analysis, we partition the range of the sending rate $X(t)$ into four different regions as shown in Fig. 3. The discontinuity in $X(t)$ when the channel rate transitions from C_0 to C_1 is clearly visible in the figure. An interesting observation is that because of the discontinuity, the sample path of $X(t)$ never resides in the region (C_0, gC_0). This can be contrasted with the window size evolution in Fig. 2 which has no discontinuities.

We use techniques from fluid analysis to develop forward differential equations for the probability density function $f_i(x, t)$. Under the assumption that $X(t)$ has a steady-state distribution in the limit $t \to \infty$, we obtain the following
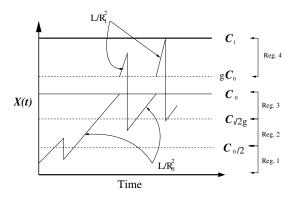
Fig. 3. Regimes for $C_1 \leq 2C_0$

system of differential equations for the distribution $f_i(x)$, where we denote $\frac{d}{dx}f_i(x)$ by $\dot{f}_i(x)$:

1) $0 < x < C_0/2$

$$\delta_0 \dot{f}_0(x) = -\gamma_0 x f_0(x) + 4\gamma_0 x f_0(2x) \qquad (1)$$
$$f_1(x) = 0$$

2) $C_0/2 < x < C_1/2g$

$$\delta_0 \dot{f}_0(x) = -\gamma_0 x f_0(x) + 4g^2 \gamma_1 x f_1(2gx) \qquad (2)$$
$$f_1(x) = 0$$

3) $C_1/2g < x < C_0$

$$\delta_0 \dot{f}_0(x) = -\gamma_0 x f_0(x) \qquad (3)$$
$$f_1(x) = 0$$

4) $gC_0 < x < C_1$

$$f_0(x) = 0$$
$$\delta_1 \dot{f}_1(x) = -\gamma_1 x f_1(x). \qquad (4)$$

The above differential equations can be solved numerically in order to obtain the rate distribution $\{f_i(t)\}$. However, the actual quantity of interest for optimization is usually the mean TCP throughput, and fortunately, an analytical expression for it can be obtained, using Mellin transforms [18]. Define ϕ_{ij} as follows

$$\phi_{ij} = \frac{\gamma_i}{\delta_j} = \frac{p_i R_j^2}{L^2}. \qquad (5)$$

Furthermore, define

$$\Delta(u) = \int_{C_0}^{C_1} e^{-\phi_{11}x^2} x^u du$$

which is the incomplete Gamma integral, and let

$$\Pi_k(u) = \frac{1}{u+2k} \prod_{i=0}^{k-1} \left(\frac{1 - 2^{-u-2i}}{u+2i} \right).$$

After some algebraic manipulation (complete details are available in [19]), we obtain the final expression for TCP throughput as:

$$\overline{X} = \frac{\Delta(2) + \sum_{k\geq 0}(\phi_{00})^k \Pi_k(2)\psi(2+2k)}{\Delta(1) + \sum_{k\geq 0}(\phi_{00})^k \Pi_k(1)\psi(1+2k)}. \qquad (6)$$

We note that though the above analytical expression may not yield itself to direct interpretation, nevertheless it is still quite useful since utilizing (6) to compute TCP throughput significantly speeds up computation time as opposed to simulations or numerical evaluation of the differential equations. Moreover, one can obtain simpler expressions for special cases, *e.g.*, $p_0 = 0$, if it is desired.

VI. NUMERICAL RESULTS

This section is devoted to quantifying the throughput gain obtainable with TCP-aware rate adaptation, verifying the accuracy of the analytical model as well as exploring the impact of the two resource contention mechanisms on multiple TCP sessions in a CDMA system.

Before evaluating the performance of the rate adaptation scheme, we must address the issue of how rate adaptation by the channel scheduler affects both channel capacity and packet error probability. Note that our framework does not assume any specific dependencies between the various channel rates and packet error probabilities, rather it assumes these are predetermined input variables.

As mentioned in Section I, rate adaptation in CDMA systems can be achieved by changing any of three variables: error coding rate, the spreading factor or modulation scheme. In this work we study the impact of the first two variables, though our framework is equally applicable to any other means of controlling the data rate. We initially assume that the channel rate and packet error are controlled *solely by changing the coding rate*. This is primarily to provide a simple basis to evaluate the accuracy of the model as well as showcase its utility in a single-user framework. Section VI-A presents the relationship we use to quantify the impact of coding rate on channel capacity and packet error probability. We use this relationship to: a) evaluate the accuracy of our model in subsection VI-B, and b) choose coding rates to maximize TCP throughput in subsection VI-C.

In subsection VI-D, we consider scenarios where resource allocation is controlled by changing the *spreading factor*. This is representative of current CDMA networks where spreading factor is the dominant control knob to adapt data rates. We evaluate the gain in TCP throughput and resultant trade-off with energy consumption as a function of the spreading factor. Finally, using the same scenario, in subsection VI-E, we evaluate a system with multiple TCP sessions, in particular the impact of α, β, number of channels *etc.* on TCP throughput.

In passing, we note that another potential control factor in wireless networks is the retransmission mechanism typically deployed at radio link layer to mitigate high frame error rates by retransmitting erroneous frames. Our model can easily incorporate the impact of link-layer retransmissions as a trade-off between channel rate and packet error probability similar to the coding rate studied here, though, it cannot account for the latency introduced by link layer retransmissions. However, extensive measurements conducted over a commercial 1xRTT network by Mattar *et al.* [2], found that the impact of rate changes of the scheduler on TCP sending rate is far more dominant, while there is little or no correlation between TCP

round trip times and the link-layer retransmissions primarily due to large RTTs and very fast re-transmissions. Hence we do not study the impact of link layer re-transmissions in this work.

A. Packet Error Probability: The Variable Coding Case

As mentioned previously, we assume that the capacity C_i and packet error probability p_i in mode i are functions of the coding rate ρ_i. Hence, for a given bit error probability, the TCP throughput is a function of the two coding rates, *i.e.*, $\overline{X}(\rho_0, \rho_1)$. The relation between the capacity and coding rate is straightforward and is given by $C_i = \rho_i \cdot C^*$ where C^* is the uncoded channel capacity.

The packet error probability however is strongly dependent on not just the coding rate, but also the coding scheme used. Consequently, one must either choose a specific coding scheme, or resort to bounds on the achievable packet error probability for a given coding rate. One such bound is the Gilbert-Varshamov [20] bound. This was used in [7] and we also use it as an approximation. The Gilbert-Varshamov bound is a bound on the parameters of a code of length B and information bit length K. It specifies that there exists a minimum Hamming distance d between any two codewords that must satisfy

$$2^B \le 2^K \sum_{j=0}^{d-1} \binom{B}{j} \qquad (7)$$

where $d-1 \le B/2$. Such a code can correct at most $t = \lfloor (d-1)/2 \rfloor$ errors. Hence, the above relation bounds the maximum number of correctable errors for any coding rate.

Let p_e denote the bit error probability for the wireless channel in consideration. Suppose that TCP packets have fixed size of L bits. The TCP packets are broken up into radio blocks of size B bits for transmission over the wireless channel. Each radio block is assumed to have K bits of information and $(B-K)$ bits for coding. Hence the coding rate is $\rho = K/B$. The packet error probability is

$$p = 1 - (1 - p_b)^{\lceil \frac{L}{\rho B} \rceil}, \qquad (8)$$

where p_b is the radio block error probability. Using the Gilbert-Varshamov bound, for a given coding rate, we can determine the maximum number of error bits, t, that are correctable using the coding scheme with rate ρ. Then,

$$p_b = \sum_{j=t+1}^{B} \binom{B}{j} p_e^j (1 - p_e)^{B-j}. \qquad (9)$$

B. Adaptive Coding Evaluation

We begin our evaluation of the TCP model by comparing its accuracy against simulations of the TCP-aware channel scheduler implemented in *ns-2*. To be concrete in this section, we set the raw channel rate C^* to 128 Kbps and the two-way propagation delay a to 200 ms. TCP packet size is set to 1024 bits. A TCP packet is divided into radio blocks of size 256 bits for transmission over the wireless channel.

The same parameters were also used for the *ns-2* simulation. TCP Reno was chosen (since the model fits that version best) and packet errors were assumed to be independent and identically distributed[7]. Since we do not account for time-outs in our model, we only simulated scenarios with large window size that result in few timeouts. Indeed, a previous study ([3]) has recommended the use of large window sizes in cellular networks to precisely avoid such time-outs due to bandwidth oscillations. The duration of each simulation run was 300 seconds. The bit error probability was held constant for the duration of the simulation, which is true under perfect power control. Hence the packet error probability is solely a function of the coding rate. We ran 20 simulations with different random seeds for each data point and the results reported are in the 95% confidence interval.

It is worth mentioning that we limited ourselves to low packet error probabilities in all the scenarios. The reasons for this are two fold. First, TCP is known to perform well only for low packet errors (less than 5%) and hence it represents the region of interest. The second reason has to do with modeling the packet loss process as an inhomogeneous Poisson process which is reasonable only for low packet error probabilities. In all the scenarios considered in this paper, the model matches the simulation results closely with errors typically less than 5% (please see [17] for error plots).

To demonstrate the importance of capturing the correlation between TCP and the scheduler as well as the presence of two states, we plot in Fig. 4 the difference between the two-rate model (which we have shown above to be accurate) and a *single-rate* model that takes only one set of parameters due to coding rate ρ_0 into consideration (please see [17] for the result of considering ρ_1). We observe that there exist regions where there is substantial difference in predicted throughput $(20\% - 30\%)$. Hence a single rate model may not be always feasible to tune performance in such systems.
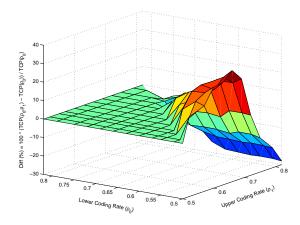

Fig. 4. Comparison between Two-Rate and Single-Rate models with parameters (C_0, p_0, R_0) for bit error probability $(p_e) = 10^{-2}$.

---

[7]Typically in wireless channels, errors are bursty affecting multiple packets. From the perspective of our model, this simply shows up as a single loss event.

TABLE I

TWO-RATE SCHEDULER: OPTIMAL ADAPTIVE CODING RATES

($C^* = 128$ KBPS).

	Analysis		Simulation	
p_e	$(\tilde{\rho}_0, \tilde{\rho}_1)$	Thr. (Kbps)	$(\tilde{\rho}_0, \tilde{\rho}_1)$	Thr.(Kbps)
10^{-1}	$0.125, 0.125$	10.85	$0.0625, 0.125$	9.22
10^{-2}	$0.630, 0.720$	64.87	$0.600, 0.690$	62.903
10^{-3}	$0.820, 0.910$	84.90	$0.820, 0.910$	81.43
10^{-4}	$0.910, 0.960$	91.99	$0.910, 0.960$	91.75
10^{-5}	$0.960, 0.990$	94.65	$0.910, 0.960$	92.43

TABLE II

TWO-RATE SCHEDULER: GAIN OF ADAPTIVE CODING.

p_e	Analysis	Simulation
10^{-1}	0%	3.8%
10^{-2}	6.9%	6.3%
10^{-3}	4.2%	3.9%
10^{-4}	5.3%	7.15%
10^{-5}	2.7%	0%

C. Optimal Coding Rates

We next turn our attention to determining the coding rates that maximize TCP throughput.

1) Two-Rate Scheduler: If we assume the *adaptive coding* case, channel rates are controlled through the coding rate. Therefore, the scheduler is essentially switching between two coding rates ρ_0 and ρ_1.

Clearly, the particular choice of ρ_0 and ρ_1 affects the achieved throughput. Intuitively, if packet error probability is close to 0 then increasing the coding rate increases throughput as well. The reason is that, in this situation, channel rate increases as a linear function of the coding rate whereas the packet error probability remains negligible resulting in increased throughput. However, if the packet error probability is large, then increasing the coding rate decreases throughput and eventually reduces it to 0. Consequently, one expects the existence of a coding rate that maximizes TCP throughput. Such behavior has been previously observed for the *single rate* case in [7] for finite capacity and [4], [5] for infinite capacity models.

We extend these previous results by identifying the optimal *pair of coding rates* within the framework of an adaptive TCP-aware channel scheduler. Table I presents the pair of optimal coding rates $\tilde{\rho}_0$ and $\tilde{\rho}_1$ obtained from both, the model and simulations, that maximize the TCP throughput $\overline{X}(\rho_o, \rho_1)$ and the corresponding throughput for different target bit error probabilities. Observe that in most cases, there is a close match between the optimal coding rates (and corresponding throughput) obtained from the model and those from simulations. In order to quantify the benefits of using two coding rates, Table II presents the relative increase in throughput as a consequence of using adaptive coding compared to a single coding rate for each bit error probability. The gain factor is defined as

$$\text{gain} = 100 \times \frac{\overline{X}(\tilde{\rho}_0, \tilde{\rho}_1) - \overline{X}(\tilde{\rho})}{\overline{X}(\tilde{\rho})} \quad (10)$$

where $\tilde{\rho}$ represents the optimal coding rate for the single rate case. Hence, for each bit error probability, the gain factor was computed by comparing throughput at the *optimal* coding rates for adaptive and static coding respectively. In all the cases, we see that the optimal throughput with adaptive coding is *at least* equal to that of static coding and higher in several situations, with improvements of up to 7%.

Intuitively, the reason adaptive coding yields a gain in throughput over static coding even with the same target bit error probability is that the channel scheduler with adaptive coding exploits knowledge of TCP sending rate. Specifically, when TCP has a small window, it sends at a low rate. Hence, the channel scheduler can offer a smaller rate to the source but with a lower error probability. As the rate increases, the scheduler switches to a higher rate to cope with TCP. This is not possible in the single rate case.

2) Three-Rate Scheduler: Next, we look at how much additional throughput gain is possible when the wireless scheduler can transition between *three* channel rates. In our current evaluation framework this essentially means that the scheduler is switching between three coding rates ρ_0, ρ_1 and ρ_2. The three-rate TCP-aware scheduler with channel rates (C_0, C_1, C_2) that correspond to coding rates (ρ_0, ρ_1, ρ_2) operates as follows:

$$C(t) = \begin{cases} C_0 & \text{if } X(t) \leq C_0 \\ C_1 & \text{if } C_0 < X(t) \leq C_1 \\ C_2 & \text{otherwise} \end{cases} \quad (11)$$

where $C(t)$ and $X(t)$ denote the channel rate allocated to the TCP session and TCP sending rate at time t, respectively.

It is worth mentioning that the *ns-2* simulations to identify the optimal rate configuration of the adaptive coding scheme is considerably time consuming even for the simpler two-rate scheduler. Computing TCP throughput via *ns-2* for each combination of rates takes almost a day for 20 runs of simulations. Therefore, we have only simulated a three-rate scheduler in this paper. However, we expect to see similar results for channel schedulers with more than three channel rates.

TABLE III

THREE-RATE SCHEDULER: OPTIMAL ADAPTIVE CODING RATES

($C^* = 128$ KBPS).

p_e	$(\tilde{\rho}_0, \tilde{\rho}_1, \tilde{\rho}_2)$	Thr. (Kbps)	Gain
10^{-1}	$0.03125, 0.06250, 0.125$	8.81	0%
10^{-2}	$0.600, 0.690, 0.720$	64.25	8.5%
10^{-3}	$0.790, 0.870, 0.910$	82.54	5.33%
10^{-4}	$0.870, 0.910, 0.960$	91.67	7.06%
10^{-5}	$0.910, 0.960, 0.960$	92.43	0%

Table III summarizes simulation results for three-rate scheduler for different BERs. In the table, $(\tilde{\rho}_0, \tilde{\rho}_1, \tilde{\rho}_2)$ denotes the optimal coding rates corresponding to three channel rates used by the scheduler that maximize TCP throughput. We have also computed the gain factor representing the percentage of throughput improvement compared to static coding throughput as defined before.

An interesting observation is that for all simulated cases the throughput gain of three-rate scheduler is roughly equal to that of the two-rate scheduler considering the margin of simulation error. In other words, most of the throughput improvement is already achieved by using only two rates and increasing the number of channel rates does not further improve the throughput.

D. Optimal Processing Gain

In the previous subsections, we assumed that the trade-off between the channel capacity and frame error rate was controlled by the coding rate. Next, we study scenarios that are more representative of current CDMA cellular networks. In CDMA networks, the dominant RF control variable that governs this trade-off is the *spreading factor*. The spreading factor is defined as the ratio of the CDMA chip rate to the actual *data rate* and relates the channel capacity and the bit error probability in the following manner. Let W denote the *chip rate* of the CDMA system and r_i the spreading factor allocated in mode i. Then the data rate C_i and *Bit Energy to Noise Ratio* are given by:

$$C_i = \frac{W}{r_i}, \qquad \frac{E_b^i}{I_0} = \frac{r_i \cdot E_c}{I_0},$$

where E_c is the *chip energy* of the data signal and I_0 the wide-spectrum interference. It should be clear from the above relations that as we *decrease* r_i, the channel capacity C_i increases, but the SINR, which directly affects the bit error probability decreases.

Typically, the bit error probability is extremely sensitive to SINR and hence, in practice, some amount of power control is required to prevent a steep rise in the error probability when the spreading factor is decreased. Specifically, the chip energy is *boosted* by a small factor that is a function of the spreading rate to prevent a sharp drop in the quality of service. Consequently, the relationship between the SINR and the spreading factor r_i is modified slightly to:

$$\frac{E_b^i}{N_0} = \frac{E_c \cdot r_i E(r_i)}{I_0}, \tag{12}$$

where $E(r_i)$ is a decreasing function of r_i and represents the factor by which the original chip energy E_c is boosted. Hence, in addition to the spreading factor, one can also control the bit error probability by an appropriate choice of the function $E(r)$, which we shall denote as the *energy profile*.

We now demonstrate how our model can be used to study the impact of rate adaptation in such an environment. Since signal power is also a resource in the above framework, we must look at both, the mean TCP throughput as well as *energy consumption*. To quantify the latter, we use the notion of *normalized energy consumption*. It represents the average energy spent per bit as a function of TCP throughput and is computed as follows. The energy spent per bit in mode i is given by $E_b^i = E_c \cdot (r_i E(r_i))$. Further, assume that the mobile session spends a fraction of time π_i in mode i and achieves an average throughput $\overline{X}_i$ in that mode. Then the normalized energy consumption is given by:

$$E_{norm} = \frac{1}{\overline{X}} \sum_i \pi_i \overline{X}_i \cdot E_b^i = \frac{1}{\overline{X}} \sum_i \widehat{f}_i(2) E_b^i$$

$$= E_c \frac{1}{\overline{X}} \cdot \sum_i \widehat{f}_i(2) r_i E(r_i). \tag{13}$$

In our experiments, we set the chip rate W to 1.2288 Mchips/sec and the basic pilot RSSI (E_c/I_0) to -7 dB based on CDMA2000 standards. We assume QPSK modulation (also a CDMA2000 standard) and a fixed coding rate of 0.6. The spreading factor r_i was allowed to take values from the set $\{4, 12, 16\}$, which gives a rate set of $\{76.8, 102.4, 153.6\}$ Kbps. We note that 78.6 Kbps and 153.6 Kbps are the two highest data rates available in CDMA2000 1xRTT with Radio Configuration Type 3 [1]. We tested different *energy profiles*, $E(r)$, and present results for two of them, denoted by E_1 and E_2.

1) Two-Rate Scheduler: Table IV presents the throughputs obtained with different rate combinations for the two-rate case under energy profile E_1 (we omit detailed E_2 results due to space constraints.). The first column in each table represents the channel rates (and hence spreading factors) used. The last column presents the energy consumption for each configuration, measured in multiples of E_c. The *optimal configuration* of spreading factors, which was chosen based on the *maximum throughput*, is marked in bold for both the single and the two rate cases. The gain in TCP throughput and difference in energy consumption between the adaptive and static cases, under their respective *optimal configurations*, are shown in Table V for both energy profiles. The energy savings is computed as:

$$\text{Energy Savings} = 100 \times \frac{E_{norm}^{static} - E_{norm}^{adaptive}}{E_{norm}^{static}}.$$

There are a number of interesting observations regarding the results. From Table V, the analytical model indicates that a small increase in energy consumption, on the order of $3\% - 4\%$, in conjunction with a two rate strategy boosts TCP throughput by $10\% - 15\%$ as compared to the single rate scenario even though the allowed *peak* channel capacities are the same. The rate gain and energy savings predictions of the simulation are comparable to those of the model for E_1 and E_2. Interestingly, simulation predicts higher benefits than the model for energy savings in the case of E_2. In fact, it predicts a *positive* energy savings of 4.2%.

The differences in energy prediction for E_2 are due to the model and simulation choosing different *optimal configurations* in the single rate case, while yielding the same configuration in the two rate case. Specifically the model overestimated (by about 9%) the TCP throughput in the single rate case for the highest rate configuration of 153.6 Kbps causing it to pick that as the optimal. In comparison the simulation chose the 102.4 Kbps rate which has higher normalized energy consumption.

2) Three-Rate Scheduler: We have also simulated a three-rate scheduler that utilizes all three rates $\{76.8, 102.4, 153.6\}$ Kbps and the same energy profiles used in the previous subsection. Table VI summarizes the results for throughput gain

TABLE IV

TWO-RATE SCHEDULER: SPREADING FACTORS ($C^* = 153.6$ KBPS) ENERGY PROFILE E_1.

(C_0, C_1) Kbps	Analysis		Simulation	
	Thr. (Kbps)	Energy Consumption ($\cdot E_c$ J/bit)	Thr. (Kbps)	Energy Consumption ($\cdot E_c$ J/bit)
$76.8, 102.4$	74.545	13.77	74.350	13.77
$76.8, 153.6$	89.83	13.08	86.793	13.12
$102.4, 153.6$	**98.180**	**13.26**	**95.316**	**13.284**

TABLE V

TWO-RATE SCHEDULER: MAX RATE CONFIGURATION.

Energy Profile	Analysis		Simulation	
	Rate Gain	Eng. Sav.	Rate Gain	Eng. Sav.
E_1	10.8%	-3.5%	14.95%	-3.5%
E_2	15.8%	-4.8%	20.5%	4.2%

TABLE VI

THREE-RATE SCHEDULER: SPREADING FACTORS ($C^* = 153.6$ KBPS).

Energy Profile	Max Thr. (Kbps)	Thr. Gain	Energy Sav.
E_1	95.209	14.8%	-3.8%
E_2	93.147	20.03%	4.2%

using the three-rate scheduler. Comparison with the throughput and energy gains for the two-rate system in Table V, we see that they are almost the same, indicating that two channel rates yields a large fraction of the benefits.

E. Multi-User Scheduler Performance

In previous subsections, we showed that by exploiting dynamic rate adaptation, it is possible to improve system throughput by 10% to 20% for a single TCP session. We next explore the impact of rate adaptation when multiple TCP sessions are present by studying the performance of the two proposed high-rate channel allocation algorithms: $\alpha-$EMPT and $\beta-$AWARE.

1) $\alpha-$EMPT Performance Evaluation: Our first set of results pertain to the simpler $\alpha-$EMPT scheduler (a brief preliminary analysis appeared in [21]). For purposes of our evaluation we utilize the same scenario as in subsection VI-D for a CDMA system. Specifically, we assume that rate adaptation is carried out by varying processing gain. Channel error probabilities were determined using the energy profile E_1 to be $\{0, 7.7, 8.7, 43.2\} \times 10^{-4}$ for channel rates of $\{9.6, 76.8, 102.4, 53.6\}$ Kbps respectively.

Fig. 5 plots the TCP throughput as a function of α and the number of supplemental channels K for $N = 10$ users. The supplemental channel rate was set to 153.6 Kbps in Figs. 5(a), 5(b) and 5(c), and 102.4 Kbps in Fig. 5(d). The figures show the evolution of throughput for fundamental channel rates of 102.4, 76.8 and 9.6 Kbps.

These three scenarios provide several interesting observations regarding the role of α, the channel rates C_0, C_1 and the number of supplemental channels K, that help explain the behavior of the $\alpha-$EMPT scheduler:

1) For a fixed α, the TCP throughput is obviously an increasing function of K. However, the throughput gain diminishes with increasing K. This can be directly attributed to the statistical multiplexing gain. Application of the single session analytical model developed in Section V reveals that the supplemental channel utilization is around 45% and 70% when the fundamental channel rates are 102.4 and 76.8 Kbps respectively. This indicates that the high-rate channel is not continuously occupied, especially in the first case, and hence can be shared to a certain extent among multiple sessions. Consequently, initial increments of supplemental channels (from $K = 1$) yield almost proportional increments in throughput because it reduces contention. However, throughput saturation sets in once K is large enough to satisfy the high-rate utilization requirements: smaller the single session high-rate utilization, faster the saturation and hence fewer channels required. For example, when the fundamental channel rate is 102.4 Kbps (Fig.5(a)), almost all throughput gain is obtained with about 5 supplemental channels since TCP sessions utilize it at most 45% of the time. We shall explore this further in subsection VI-E.3.

2) For small values of K, the throughput is sensitive to the choice of α and its behavior depends on the fundamental *and* supplemental channel rate combination. As K increases, the throughput becomes less sensitive to α since supplemental channels are increasingly available when requested and the contention likelihood decreases.

It is worthwhile further exploring the detailed behavior of TCP throughput as a function of α for small values of K. For ease of exposition, consider $K = 1$ in a low-loss system, *i.e.*, $p_1 \rightarrow 0$[8]. When $\alpha = 0$, a TCP session that is granted a supplemental channel can fully utilize the high-rate channel (under the low-loss assumption) till it experiences congestion (when TCP transmission rate hits C_1). However, this can also lead to a lower likelihood of occupancy of the high-rate channel (*i.e.*, it stays empty) since another session may not be available to exploit the available channel when the current session relinquishes it (see Fig. 6).

When $\alpha = 1$, the supplemental channel occupancy increases (assuming sufficient requests) since the release of the high rate channel is due to preemption. However, this can result in lower *efficiency* compared to $\alpha = 0$ if the supplemental channel rate is large, since the session may get preempted before it can sufficiently increase its transmission rate (see Fig. 6). To summarize, the trade-off between these two factors, which in

---

[8]Intuitively, as loss rate becomes high, performance becomes insensitive to α since a session frequently experiences channel error in the supplemental channel.
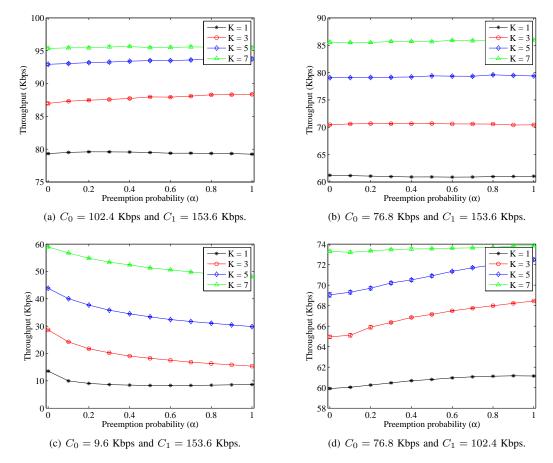
(a) $C_0 = 102.4$ Kbps and $C_1 = 153.6$ Kbps.

(b) $C_0 = 76.8$ Kbps and $C_1 = 153.6$ Kbps.

(c) $C_0 = 9.6$ Kbps and $C_1 = 153.6$ Kbps.

(d) $C_0 = 76.8$ Kbps and $C_1 = 102.4$ Kbps.

Fig. 5.   Impact of α on TCP throughput for $N = 10$.

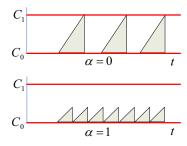turn is determined by the relative values of C_0 and C_1 governs the TCP throughput as a function of α.

Fig. 6.   Impact of α on supplemental channel utilization.

This behavior is exemplified in Fig. 5. When $C_0 \ll C_1$, (*e.g.*, $[9.6, 153.6]$ Kbps in Fig. 5(c)), the high-rate channel *utilization* dominates since a session in the high-rate channel can gain significant increase in throughput compared to the fundamental channel. Furthermore, since the fundamental channel is small, request rate for the high-rate channel is large enough to avoid any potential loss of occupancy. Consequently, $\alpha = 0$ is optimal because it allows a session to fully utilize the channel. As C_0 increases relative to C_1, the gain in throughput from utilization of the high-rate channel diminishes (since C_0 is already large) and request rates reduce (because sessions spend more time in C_0 as described by our model earlier). Hence, the high-rate channel occupancy factor becomes more

important. Figs. 5(a) and 5(d) illustrate this scenario. Here $2C_0 > C_1$, and $\alpha = 1$ maximizes occupancy, which yields the optimal throughput. An intermediate case is $2C_0 = C_1$ (Fig. 5(b)), where the two factors balance each other. This results in a TCP throughput *insensitive* to α.

Next, we look at the impact of α on TCP throughput gain. In Fig. 7, we have plotted the maximum throughput gain of $\alpha-$EMPT across different channel configurations for $N = 10$ users and various supplemental channels K. The gain was computed (over the entire range of α) as:

$$\text{Max gain} = 100 \times \frac{\max_\alpha \overline{X}_{\alpha-\text{EMPT}} - \min_\alpha \overline{X}_{\alpha-\text{EMPT}}}{\min_\alpha \overline{X}_{\alpha-\text{EMPT}}}$$

One can clearly see that the $\alpha-$EMPT scheduler, and in particular the value of α is influential when the gap between C_0 and C_1 is very large ($C_0 = 9.6$ and $C_1 = 153.6$ Kbps), resulting in throughput gains of $60\% - 90\%$ by appropriately tuning α. However, when C_0 is closer to C_1 ($\geq C_1/2$), performance is almost insensitive to α with small gains of about $2\% - 5\%$.

2) $\beta-$AWARE Performance Evaluation: Given the relative insensitivity of TCP throughput to the $\alpha-$EMPT scheduler when the gap between C_0 and C_1 is not large, we next turn to the more sophisticated scheduler $\beta-$AWARE.

We studied the impact of the $\beta-$AWARE scheduler on TCP throughput as a function of β and found it to have a similar dependency as the $\alpha-$EMPT scheduler on the gap between
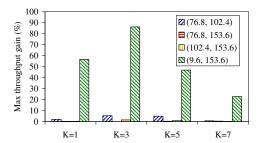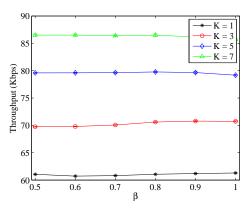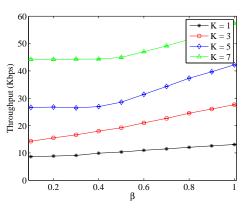
Fig. 7. Impact of α across all scenarios.

Fig. 9. Comparison of $\alpha-$EMPT vs $\beta-$AWARE scheduler.

C_0 and C_1 as well as number of supplemental channels. This results are shown in Fig. 8.

(a) $C_0 = 76.8$ Kbps and $C_1 = 153.6$ Kbps.

(b) $C_0 = 9.6$ Kbps and $C_1 = 153.6$ Kbps.

Fig. 8. Impact of β on TCP throughput for $N = 10$.

Fig. 9 presents a comparison in performance between $\beta-$AWARE and $\alpha-$EMPT schedulers across different rate configurations, number of users and supplemental channels (the numbers in brackets represent (N, K)). It shows the gain achieved in throughput with the *best* configuration of the $\beta-$AWARE scheduler through appropriate selection of β when compared against the *best* configuration of the $\alpha-$EMPT scheduler. In other words, the gain was computed as:

$$\text{Max gain} = 100 \times \frac{\max_\beta \overline{X}_{\beta-\text{AWARE}} - \max_\alpha \overline{X}_{\alpha-\text{EMPT}}}{\max_\alpha \overline{X}_{\alpha-\text{EMPT}}}$$

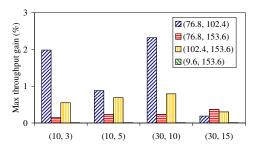Somewhat surprisingly, we observe that in all scenarios the $\beta-$AWARE scheduler yields little benefit over the $\alpha-$EMPT
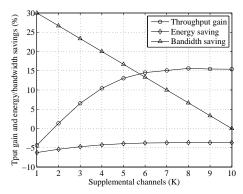
scheduler, with about $2\% - 5\%$ improvement. We note that by tuning β and α we cover a large range of potential behaviors. This leads us to conjecture that TCP throughput in such situations is relatively insensitive to channel allocation strategy when C_0 and C_1 do not differ by a large value and most of the gain in throughput is achieved through rate adaptation.

3) Multiplexing Gain with Multiple Flows: This leads us to the final aspect of the study of benefits of rate-adaptation in TCP multi-session systems, specifically resource efficiency. As was alluded to earlier when discussing the $\alpha-$EMPT scheduler, results from the single session analytical model indicate that in a multi-rate system, the high-rate channel is not fully utilized. This can lead to significant savings. In CDMA systems, high rate channels are achieved by reducing the orthogonal Walsh code length. Since they are highly susceptible to interference, it limits the number of simultaneous high-rate channels that can be supported (the higher the rate, the fewer the channels), which in turn affects capacity. Consequently, it is desirable to provide throughput gains while at the same time efficiently utilizing high-rate channels.
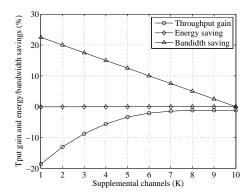
By exploiting the statistical multiplexing gain inherent in rate adaptation, we demonstrate that one can achieve similar or higher throughput as with single rate channels but with a lower total bandwidth.

Fig. 10 presents the throughput and bandwidth savings when using multi-rate channels in a multi-user system compared to single-rate channel allocation for the two different energy profiles E_1 and E_2. For purposes of comparison, the channel rate in the single-rate case was chosen to be the one that yields the maximum throughput as computed in Section VI-D. To demonstrate the impact of rate adaptation, for the multi-rate case, we deliberately chose rates such that the fundamental channel rate was *lower* than the single rate channel allocation even if it was sub-optimal in the sense that it did not yield the maximum throughput computed in Section VI-D.

The total bandwidth consumed in the single rate case was given by NC, where C is the single channel rate and by $(N - K)C_0 + KC_1$ in the multi-rate case. Fig. 10 plots the percentage difference for achieved throughput and consumed bandwidth between the multi-rate and single-rate cases. As can be clearly seen, for energy profile E_1, the multi-rate system consumes less bandwidth and yields higher throughput gains of about 15% albeit at the cost of a small increase in energy. With energy profile E_2, when $K \geq 6$, the multi-rate system yields almost the same throughput as the single-rate system but consumes about $10\% - 15\%$ less bandwidth. The figures

(a) Profile E_1: $C = 153.6$, ($C_0 = 102.4, C_1 = 153.6$) Kbps.

(b) Profile E_2: $C = 102.4$, ($C_0 = 76.8, C_1 = 153.6$) Kbps.

Fig. 10.   Throughput gain and energy saving as function of K.

also clearly indicate the benefit of the multiplexing gain, in that the gains saturate around $K = 5$ channels for energy profile E_1 and $K = 7$ channels for E_2, *i.e.*, performance is close to that achieved with $K = N$ channels.

VII. Conclusion

In this paper we proposed techniques for optimizing the rate adaptation feature of modern CDMA systems in a TCP-aware fashion. Our proposed system adapts its channel rate in response to the TCP sending rate, allowing it to trade-off channel rate and FER in a TCP-friendly manner.

For the single TCP session case, we developed an analytical model for a *two-state* system that explicitly accounts for the interaction between our proposed scheduler and TCP dynamics as well as the presence of two distinct regimes in terms of channel rate, packet error probability and round trip time. The model was utilized to show how to select either error coding or processing gain in optimizing the rate adaptation scheme to improve TCP throughput. Improvements of 15% to 20% were observed compared to a system that does not exploit rate adaptation. Furthermore, we also evaluated systems that use *three* channel rates via simulations and observed that they yield no significant gains compared to a two-rate system.

Finally, we extended the rate adaptation framework to multiple TCP sessions in a CDMA system. In such an environment, the number of high rate channels is less than the number of active users and thus the high rate channels must be shared in a fair fashion. We proposed two channel allocation policies

based on preemption and presented extensive simulation results on their performance. Our results indicated that the rate adaptation feature with selected rate combinations, rather than the channel allocation policy, is affecting the system throughput. We also showed that by exploiting statistical multiplexing gain, a system with just a few high-rate channels achieves per-session throughput close to that of a single session system with potential savings in both bandwidth and energy budget.

References

[1] Telecommunications Industry Association, "TIA EIA IS-2000," March 2000. [Online]. Available: www.tiaonline.org/standards/sfg/imt2k/cdma2000/

[2] K. Mattar, A. Sridharan, H. Zang, I. Matta, and A. Bestavros, "TCP over CDMA2000 networks: A cross-layer measurement study," in *Proc. PAM*, Louvain-la-neuve, Belgium, April 2007.

[3] E. Chaponniere, S. Kandukuri, and W. Hamdy, "Effect of physical layer bandwidth variation on TCP performance in CDMA2000," in *Proc. IEEE VTC Spring*, April 2003, pp. 336–342.

[4] C. Barakat and E. Altman, "Bandwidth tradeoff between TCP and link-level FEC," *Computer Networks*, vol. 39, no. 5, pp. 133–150, 2002.

[5] B. Liu, D. L. Goeckel, and D. Towsley, "TCP-cognizant adaptive forward error correction in wireless networks," in *Proc. IEEE Globecom*, Taipei, Taiwan, November 2002.

[6] D. Barman, I. Matta, E. Altman, and R. E. Azouzi, "TCP optimization through FEC, ARQ and transmission power trade offs," in *Proc. WWIC*, Frankfurt, Germany, February 2004.

[7] F. Baccelli, R. Cruz, and A. Nucci, "CDMA channel parameters maximizing TCP throughput," in *Proc. Workshop on Information Theory and its Applications*, La Jolla, USA, February 2006.

[8] A. Chockalingam, E. Altman, J. V. K. Murthy, and R. Kumar, "Cross-layer design for optimizing TCP performance," in *Proc. IEEE ICC*, Seoul, Korea, May 2005.

[9] M. Zorzi, A. Chockalingam, and R. R. Rao, "Throughput analysis of TCP on channels with memory," *IEEE J. Select. Areas Commun.*, vol. 18, no. 7, pp. 1289–1300, 2000.

[10] A. Chockalingam and G. Bao, "Performance of TCP/RLP protocol stack on correlated rayleigh fading DS-CDMA links," *IEEE Trans. Veh. Technol.*, vol. 49, no. 1, pp. 28–33, 2000.

[11] K. L. Gray and D. L. Noneaker, "The effect of adaptive-rate coding on TCP performance in wireless communications," in *Proc. EUROCOMM*, 2000.

[12] E. Altman, C. Barakat, and V. M. R. Ramos, "Analysis of AIMD protocols over paths with variable delay," in *Proc. IEEE INFOCOM*, Hong Kong, March 2004.

[13] M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in *Proc. ACM MOBICOM*, 2002, pp. 71–82.

[14] J. P. Singh, Y. Li, and N. Bambos, "Channel state awareness based transmission power adaptation for efficient TCP dynamics in wireless networks," in *Proc. IEEE ICC*, Seoul, Korea, May 2005.

[15] W. W. an Chun Zhang, H. Zang, J. Kurose, and D. Towsley, "Inference and evaluation of split-connection approaches in cellular data networks," in *Proc. PAM*, Adelaide, Australia, March 2006.

[16] V. Misra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, August 2000, pp. 151–160.

[17] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley, and R. Cruz, "Tcp-aware resource allocation in CDMA networks," Sprint Advanced Technology Labs," Technical Report *RR06-ATL-030566*, March 2006.

[18] F. Baccelli and K. B. Kim, "TCP throughput analysis under transmission error and congestion losses," in *Proc. IEEE INFOCOM*, Hong Kong, March 2004.

[19] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley, and R. Cruz, "TCP-aware resource allocation in CDMA networks," in *Proc. ACM MOBICOM*, Los Angeles, USA, September 2006, pp. 215–226.

[20] F. J. McWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*.   New York, USA: North-Holland, 1977.

[21] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley, and R. Cruz, "Modeling TCP in a multi-rate multi-user CDMA system," in *Proc. Networking*, Atlanta, USA, May 2007.