TCP-Aware Power Control in Wireless Networks

Hui Zang[†], Majid Ghaderi[††], and Ashwin Sridharan[†]
[†] Applied Research, Sprint Nextel, {hui.zang,ashwin.sridharan}@sprint.com
[††] Department of Computer Science, University of Calgary, mghaderi@ucalgary.ca

***Abstract*—Modern cellular networks commonly deploy rapid channel rate adaptation to vary the wireless capacity in response to channel conditions while maintaining a fixed target error rate (typically 1%). Although desirable in terms of throughput for inelastic applications that do not adapt to network conditions, a low fixed target error rate incurs the expense of significant power consumption, especially at high transmission rates. In this work, we show that elastic traffic, in particular TCP, benefits greatly from the perspective of power efficiency when we also incorporate** *target error rate adaptation*. **More specifically, TCP behavior, although sensitive to packet errors, is not uniformly so. When TCP has a small window, it requires extremely low packet error rates. However, for large windows, especially with a buffer, TCP can tolerate larger loss rates. The contribution of this work is in conducting a detailed and realistic investigation into how beneficial target error rate adaptation is for TCP in terms of reducing power and impact on throughput. Our work differs from past contributions in that we explicitly take into account the impact of the buffer and a variable channel. We devise simple local power-adaptation policies based on TCP behavior and study them with the help of a numerical model. Finally, we present a detailed investigation of our policies using actual modulation schemes and real channel traces collected on a commercial 1xEV-DO network. The results show that compared to the existing scheme, our policies save typically about 20% to 30% power with marginal or no reduction in throughput.**

I. INTRODUCTION

Rapid, sub-second channel aware rate adaptation has become a *de facto* feature on modern third generation [1] and upcoming fourth generation [2] wireless networks. The MAC and physical layers on such networks support a pre-determined set of coding and modulation schemes, which translate into a discrete set of rates. One example is CDMA 1xEV-DO that applies a channel-aware transmission policy on the downlink (from base station to device) in a slot-by-slot fashion. In each time-slot, the mobile device rapidly senses its Signal to Interference-and-Noise Ratio (SINR), and selects the highest suitable transmission rate to receive data that can support an *a priori* specified target frame error rate (FER). Hence, the channel rate fluctuates from slot to slot in response to channel variations while sustaining the target FER.

A key aspect of such a system is that the target FER is typically fixed to a low value (*e.g.*,1% in 1xEV-DO). In order to achieve this low FER, data transmissions require a reasonably large amount of power, especially at high data rates (depending on channel conditions). One of the main reasons the target FER is chosen to be low, is because TCP, the dominant transport protocol is known to require extremely low packet error rates for good performance [3].

However, as we show in this paper, the elastic nature of TCP *in conjunction* with network configurations commonly adopted by cellular networks operators translates into an impact of packet loss on perceived user throughput that is *not uniformly* severe over time. In other words, there are times when a packet loss does not affect the TCP throughput, and hence the transmission power can be reduced without sacrificing the throughput, leading to improved power efficiency. We elaborate more on this with respect to TCP dynamics.

TCP[1] is a window-based transport protocol that dynamically probes the network for available bandwidth by slowly increasing the window of transmitted and unacknowledged packets and reacts to a packet loss by reducing its window by half. Hence, any packet loss has a severe impact on TCP window size, and possibly TCP throughput. The impact to throughput is most severe at small windows, when TCP has a higher likelihood of time-outs. However, TCP window size need not always translate into TCP/application throughput. Specifically, due to large rate variations on modern wireless channels, network operators typically deploy *large per-user buffers* at the base-station to absorb transient deviations between TCP transmission rate and network capacity. Hence, when TCP has a large window, it builds large queues and the *instantaneous* TCP throughput is the same as that of the wireless channel. In such scenarios, a packet loss only results in temporary reduction in queue length, which if large enough to begin with, would mitigate instantaneous throughput degradation to much less than 50%. Hence, in this regime, TCP is potentially far more tolerant to channel errors, allowing transmission with reduced power.

The objective of this paper is to demonstrate that substantial power savings (20% to 30%) can be obtained on the downlink with marginal or even no loss in long term TCP throughput through simple schemes that exploit this aspect of TCP dynamics as well as the presence of a buffer and a variable channel. In particular, we show that the *target FER* can be appropriately increased based on TCP state, which has a direct impact on the amount of required transmission power, with minor reductions in throughput. Apart from the obvious advantage of increased energy efficiency, reduction in power also has two added benefits:

1) Frequency re-use is common in cellular networks which causes inter-cell/sector interference. Minimization of transmission power reduces interference resulting in improved network capacity.
2) In future OFDM networks such as WiMAX networks, transmission power is divided across multiple users scheduled in each slot. Reduction of required per-user

---

[1] We consider TCP NewReno in this paper as it is the most widely used version of TCP in the Internet.

power increases the network capacity by allowing more users to be scheduled (or fewer at higher rates).

There is a large body of work directly related to optimization of the target FER as a function of TCP behavior (see Section II for a detailed comparison). However, most of these studies aim at improving TCP throughput rather than studying the impact of power, and also do consider the impact of a buffer or a variable channel.. The closest work in spirit to ours is [4] which also studies the problem of power-optimization with respect to TCP and proposes a complex dynamic programming based solution. However, they also do not consider the impact of the buffer. Our contributions can be summarized as follows:

1) We propose a TCP-aware power control mechanism that exploits the presence of large buffers in wireless networks to improve *power efficiency.*
2) We develop simple models to illustrate the impact of the buffer on TCP throughput, and how it affects the required target FER. Utilizing these models, we compose simple local heuristics to control the trade-off between application throughput and the transmission power. We study this relation in detail utilizing a simple numerical model and consequently provide *trade-off curves* between required power and TCP throughput.
3) Finally, we evaluate the impact of these policies using *ns-2* simulations. The evaluation is carried out with actual channel traces collected from a commercial 1xEV-DO wireless network under single-user as well as multi-user and multi-cell scenarios.

The remainder of the work is organized as follows. Section II compares this work with existing related work. Section III illustrates how power affects instantaneous throughput in the presence of a buffer and channel fluctuations using simple models. This serves as the basis for our TCP-aware power adaptation mechanisms and a number of heuristics in Section V. A detailed evaluation of the impact of our power adaptation policies on power efficiency and throughput is presented in Section V. Future directions are outlined in Section VI.

II. RELATED WORK

There is extensive literature related to optimization of TCP performance on wireless channels. One class of work, which could be termed *TCP enhancement*, either introduces end-to-end TCP modifications or *split* the TCP connection with the help of an intelligent agent. A few examples of the former are TCP Westwood [5], TCP-Freeze [6] and the Eifel timer [7]. Examples of the latter are Snoop [8] and the ACK [9] and Window regulator [10]. Detailed survey of such techniques is presented in [11] and [12].

The approach presented in this work is a *link-layer* optimization approach that adapts the RF layer to TCP dynamics rather than the other way around. In this view, our work is closer in philosophy to previous literature that optimizes link layer parameters like Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ) to improve TCP.

Identification of a single set of *optimal* link-layer parameters to maximize TCP throughput has been carried out in many past studies. References [13] and [14] showed that there exists a coding rate that maximizes TCP throughput. Baccelli *et al.* [15] conducted a similar study but with respect to determining the optimal CDMA processing gain for maximizing TCP throughput. See also [16] and [17], [18] for other cross-layer optimization techniques proposed to improve TCP throughput. All these previous studies however considered a static scenario with only a *single* optimal configuration that ignored buffer and channel dynamics. Barman *et al.* [19] studied the impact of signal power and ARQ on TCP throughput and the associated trade-off. However, they also considered only a static scenario and a single instance of parameters.

Adaptive target FER computation on the fly has been studied in [14] and [20] via simulations. In both cases, however, the scheduler behavior was agnostic to TCP *state* and did not incorporate impact of a buffer or variable channel rates.

The closest related work is that of Singh *et al.* [4], where the authors studied optimization of transmission power to maximize TCP throughput. They explicitly considered TCP dynamics in the selection of the transmission power level and formulated a complex optimization program to identify the necessary power for *each* TCP state, and studied the related power-throughput trade-off. We share a similar goal and although we provide simple local heuristics, we take into account the impact of buffer and a Markovian channel, which was not considered in [4].

III. TCP-AWARE POWER ADAPTATION

This section outlines our overall approach to improving power efficiency of TCP transmission over a wireless channel. Our metric for TCP performance in this work is *long-term* TCP throughput as observed in long lived sessions, where the dominant TCP state is that of congestion-avoidance rather than slow-start. For ease of exposition, we shall first begin with the case when the channel is assumed to be static (or slowly varying). This allows us to highlight, using simple models, how changing transmission power directly impacts TCP throughput. Later, in Section IV, we shall use these results to develop algorithms for the more realistic scenarios with wireless channel variations.

In wireless networks, packet loss is directly affected by the transmission power at the base station. Round trip time, on the other hand, is indirectly affected by the transmission power by controlling the wireless channel transmission rate. It is well known that both packet losses and round trip time adversely affect TCP performance. However, TCP is far more sensitive to packet losses than round trip time [3]. Consequently, the primary impact of transmission power on TCP is through the packet-error rate achieved on the channel. Therefore, our approach is one that characterizes the impact of transmission power on TCP performance through the channel packet error rate, and utilizes it to provide insights into how power savings can be achieved without sacrificing throughput.

Ideally, in order to achieve this objective, we should obtain a relation between the instantaneous transmission power and the

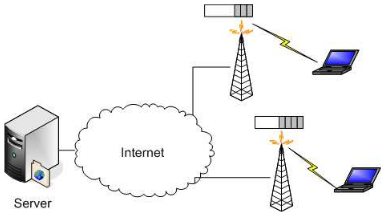


Fig. 1. Downlink of a cellular system with per-user buffering at the base station.

TCP throughput (*i.e.,* the objective *cost function*). One could then pick a policy, which maximizes the latter. However, past research has demonstrated that obtaining analytical expressions for TCP throughput in a wireless network such as the one considered in this paper is quite intractable, even in simple scenarios, *e.g.,* a single TCP session. Furthermore, the primary objective of this paper is to obtain an empirical understanding of how power affects TCP throughput rather than developing a detailed model of TCP throughput. Towards this end, we focus on a *local* objective, namely the *instantaneous change in* TCP throughput and devise simple intuitive analytical models that, albeit involving simplistic assumptions about the behavior of the system, nevertheless capture the main aspects of how transmission power affects TCP. Utilizing insight from these models, we then devise policies that improve power utilization in realistic network scenarios.

A. System Model

We focus on the wireless downlink of a single user in a cell, involved in a long-lived TCP session (*e.g.,* a large file download). Fig.1 illustrates pertinent features of the system. The base station possesses a large per-user buffer of size B to enqueue packets for transmission, and the wireless channel capacity is assumed to be C packets per second. The round trip propagation delay between the server and user is denoted by d and the round trip time (including queuing delay) by R. The downlink operates in a time slotted fashion, where each time slot is ΔT seconds. In each time slot, the base station transmits data to the user at an appropriate power level.

We focus on the evolution of system throughput in a single time slot. For tractability, we assume the absence of time-outs, buffer overflow losses, and feedback delay, aspects which we shall address in Section IV and V. Let $Y[t]$, $X[t]$, and $Q[t]$ denote the instantaneous application throughput, TCP sending rate and queue length at the base station, respectively, in time slot t. Let $W[t]$ denote the TCP window size (*i.e.,* outstanding packets) at time t. Finally, and most importantly, we define the packet error rate $p_e[t]$ to be related to the transmission power $P[t]$ as $p_e[t] = f(P[t])$, where f is a monotonically decreasing function.

B. TCP Aware Power Adaptation

We now analyze the impact of the transmission power $P[t]$ on the *instantaneous* throughput. First consider the case when the queue is not occupied, *i.e.,* $Q[t] = 0$. In such a case, the instantaneous throughput $Y[t]$ is simply the TCP transmission rate $X[t]$. We utilize the fluid model for TCP [21], wherein the TCP transmission rate is given by $X[t] = W[t]/d$, and assume that the instantaneous packet error probability can be modeled as an inhomogeneous Poisson process with rate $p_eX[t]$ [22]. The discretized *change* in throughput (ΔY) from (say) slot t to $t+1$ can be computed as:

$$\Delta Y = -p_e[t]X[t]\Delta T\frac{X[t]}{2} + (1 - p_e[t]X[t]\Delta T)\frac{\Delta T}{d^2}\,. \quad (1)$$

The first part of Eqn. (1) captures the impact of a packet error, in which case, TCP transmission rate drops by half, and the latter part captures the growth in TCP rate by amount $\Delta T/d^2$, when there is no packet error.[2] For small values of ΔT, the latter term is quite small due to the ΔT^2 term, in line with the assertion that for empty queues, throughput is reasonably insensitive to *small* changes in round trip times. Hence, we focus on the first term for the purpose of evaluating the impact of the transmission power.

Differentiating the first term in Eqn. (1) with respect to transmission power $P[t]$, we obtain:

$$\frac{\partial \Delta Y}{\partial P[t]} = -f'(P[t])\frac{X[t]^2}{2}\Delta T\,. \quad (2)$$

In order to better explain the role of Eqn. (2), it is illustrative to analyze its behavior with the help of an example function for $f(P)$. For example, for *M-ary* shift keying and an AWGN channel, the error probability and transmission power are approximately related as: $p_e \approx e^{-\alpha P^2}$, where α is some constant depending on the modulation scheme. Plugging this function in Eqn. (2), we obtain:

$$\frac{\partial \Delta Y}{\partial P[t]} = \alpha P[t]e^{-\alpha P[t]^2}X[t]^2\Delta T\,. \quad (3)$$

Clearly, as also indicated by the above equation, throughput is always going to be a non-decreasing function of power. What we wish to explore is *a regime of diminishing returns, wherein substantial increase in power yields marginal increase in throughput*. Eqn. (3) indicates that for a given TCP transmission rate $X[t]$, the *change* in TCP throughput is initially zero for small values of power P. This is because for all low values of power P, $p_e \approx 1$ and hence, TCP will almost surely experience a channel error resulting in $\Delta Y = -X[t]/2$ and hence $\frac{\partial \Delta Y}{\partial P} = 0$. As the power increases, the gain in throughput increases due to reduction of channel error. However for large values of power, the $e^{-\alpha P^2}$ term dominates and the gain in throughput once again falls rapidly with increasing power, which is the desired regime of diminishing returns. It suggests an intuitive method to characterize the trade-off between the TCP throughput and transmission power. Specifically, one could choose the *largest* power such that

$$\frac{\partial \Delta Y}{\partial P} \le \gamma, \quad (4)$$

where, γ is a parameter that indicates the gain in throughput as a function of power. Note that the relationship in Eqn. (3) also takes into account the current TCP transmission rate $X[t]$. Specifically, for larger values of TCP rates $X[t]$, it takes a larger value of power P before the gain starts to fall below γ.

---

[2] Note that we have ignored a third term accounting for both packet-loss *and* rate increase in the same slot. This is reasonable since the former event dominates anyway.

Next, let us look at the case when the queue is occupied, *i.e.*, $Q[t] > 0$. In this case, the instantaneous throughput is *not* the TCP transmission rate $X[t]$, but rather the channel capacity C. The TCP transmission rate is given by:

$$X[t] = \frac{W[t]}{R[t]} = \frac{C \cdot d + Q[t]}{R[t]}$$

where $R[t] = d + Q[t]/C$ (ignoring feedback delay). For simplicity, we approximate the TCP transmission rate as $X[t] = \frac{W[t]}{d}$, *i.e.*, we ignore the queuing delay. The impact of the queue is however captured in the numerator $W[t]$. With this approximation, the change in throughput across a time slot can be written as:

$$-p_e[t]X[t]\Delta T \max(0, C - \frac{X[t]}{2}) + (1 - p_e[t]X[t]\Delta T) \cdot 0$$

$$\Delta Y = -p_e[t]X[t]\Delta T \max(0, \frac{C - Q[t]/d}{2}) \tag{5}$$

We bring to attention the impact of a buffer in Eqn. (5) as compared to Eqn. (1). In Eqn. (1), a channel error *always* results in significant drop in throughput (half). However, in the backlogged case, if there is a channel error, which is captured by the first term, then the drop in throughput is no longer $X[t]/2$, but rather a function of the channel capacity and the queue backlog. In other words, the larger $X[t]$ is, the smaller the impact of a channel error on the throughput. At the extreme point, if $X[t] > 2C$ or $Q[t] > C \cdot d$, *i.e.*, the window size is more than twice the bandwidth delay product, then a channel error, which reduces TCP window by half, does not impact instantaneous throughput since the buffer remains backlogged till TCP completely recovers from the loss.

Similarly, if there is no channel error, then the throughput stays at C, *i.e.*, there is no change, which is captured by the latter term.[3]

Differentiating the above expression with respect to P as before, we obtain:

$$\frac{\partial \Delta Y}{\partial P} = -f'(P)X[t] \cdot \max(0, C - \frac{X[t]}{2}) \,. \tag{6}$$

Eqn. (6) captures the impact of the buffer on the amount of required power. As the queue fills up, the potential loss in throughput becomes less critical, till a point when $X[t] > 2C$, at which point a drop in TCP transmission rate has no impact on achieved throughput (since the queue remains occupied during the entire recovery period).

Eqn. (2) and Eqn. (6) can be combined as:

$$\frac{\partial \Delta Y}{\partial P} = -f'(P)X[t] \cdot \max(0, \min(X[t], C) - \frac{X[t]}{2}) \,. \tag{7}$$

Eqn. (7) in conjunction with Eqn. (4) provides a useful template for a TCP-aware power policy. Such a policy would choose a threshold level γ that reflects the desired trade-off between power efficiency and throughput. Then, in each slot we monitor the TCP transmission rate $X[t]$ (or queue state) and utilize Eqn. (7) to select the power that satisfies Eqn. (4).

In passing we note that the impact of transmission power on TCP throughput, captured in Eqn. (7) represents a simplified model of TCP dynamics where we have ignored time-outs, congestion losses and feedback delay. It however provides useful initial guidelines which we shall extend in the next section to develop power-adaptive algorithms that also address these issues.

IV. POWER ADAPTATION ALGORITHMS

The previous section outlined key characteristics of a power adaptation policy when transmitting TCP packets over a static wireless channel. Specifically, in the absence of a backlog, one must minimize packet losses. However as the backlog increases, one can be conservative in the usage of power.

We now incorporate the fact that in commercial cellular systems, the channel rate and conditions are actually *dynamic*. With minor differences, the downlink channel in modern cellular networks (*e.g.*, [1] and [2]) operates in the following fashion. In a pre-specified portion of each time-slot, the base station transmits a *pilot* signal at maximum possible power (say P_{max}). Each device associated with that station measures the SINR of the received pilot and reports it back [2], or requests the highest data rate, called DRC (Data Rate Control) that can be supported at power level P_{max} for a pre-specified and *fixed* target FER. For example, 1xEV-DO can support 12 distinct data rates. In the context of our work, we assume a mechanism similar to the former, *i.e.*, one, where each device reports the SINR, and the base station decides the rate. In typical cellular networks, once a rate has been selected, the base-station transmits to the user at the requested DRC(s) with maximum power P_{max}. We shall refer to this policy as the *baseline* power policy. Our aim is to precisely reduce the amount of transmission power in a time slot with minimal change in throughput.

A detailed characterization of the 1xEV-DO channel is carried out in [23], which shows that for mobile devices the selected DRC can change every few milli-seconds due to rapid change in channel conditions. Since typical round trip times for any connection are of the order of tens of milli-seconds, this can result in large asymmetries between source sending rate and the channel capacity. In order to absorb such fluctuations, operators usually deploy large buffers at base stations, which is one of the key elements we exploit to improve power efficiency. The size of the buffers is typically set to be larger than typical TCP maximum window size (64 KB) and hence congestion losses are minimal in such networks [23], justifying this assumption in Section III. To account for variability in channel conditions, we propose the following approaches.

A. Moving Average Algorithm

This approach is essentially a direct extension of the static scenario discussed in the previous section, with the difference that the channel capacity is continuously approximated via a *moving average* of the instantaneous channel capacity. Specifically, let $C[t]$ represent the instantaneous channel rate in slot

---

[3] We note that the above expression does not account for congestion losses due to a finite buffer. Since power control does not affect congestion losses, we do not consider them. However in later sections, our analytical model explicitly accounts for congestion losses.

t. Then, in each slot, the power adaptation algorithm works as follows:

1) The moving average estimate C is computed as follows:
$$C \leftarrow \beta C[t] + (1-\beta)C\,. \tag{8}$$
2) The TCP sending rate $X[t]$ is monitored at the base-station.
3) Finally, we use Eqn. (7) and the computed average channel rate C, along with a pre-specified threshold γ, to compute the desired power level.

As the channel fluctuates, by controlling the moving average window factor β, one can control how rapidly C reflects $C[t]$. For example, when $\beta = 1$, C equals to $C[t]$. Clearly a major benefit of such an approach is the simplicity and the fact that it requires no *a priori* knowledge of channel statistics. However, one must tune the factor β appropriately. In Section V, we show that an implementation in which $\beta = 1$, and one with $\beta = 0.01$ have distinct performance.

B. Markovian Channel State Transition Algorithm

An alternate approach to extend our power adaptation algorithm to a variable channel is through knowledge of channel statistics. As explained above, in commercial cellular networks, the channel is assigned a *discrete rate* or state in each time slot. We assume that the channel state process is Markovian (justified by studies in [23]) and that we have knowledge of the channel state transition probabilities. Under such a framework, one can then design a power policy cognizant of future channel states, which we explain in more detail next.

For purposes of simplicity, let us consider a channel that switches between just two channel states S_0 and S_1. Let $C_0(C_1)$ and $R_0(R_1)$ denote channel rate and round trip time in state $S_0(S_1)$. We assume that $C_0 < C_1$. Let p_{ij} denote the transition probability from state i to j.

We begin with the first type of transition $S_0 \rightarrow S_1$. If the queue is empty, then the channel transition results in a small increase in throughput due to reduction in transmission delay, which in turn reduces round trip time. Approximating the change in throughput with a fluid equation, wherein only one event, either packet loss *or* a channel state transition occurs, which is a reasonable assumption for very small time duration (such as a slot in 1xEV-DO networks), we obtain:

$$\Delta Y = -p_e[t]X[t]\Delta T\frac{X[t]}{2} + (1-p_e[t]X[t]\Delta T)p_{00}\frac{\Delta T}{R_0^2} + (1-p_e[t]X[t]\Delta T)p_{01}\frac{\Delta T}{R_1^2}\,. \tag{9}$$

As before, ignoring the smaller $O(\Delta T^2)$ propagation delay terms and differentiating with respect to P, we obtain the same expression as Eqn. (2), and hence a similar throughput-power relationship applies.

The more interesting case is when the queue is occupied. Again, we incorporate the impact of queue by approximating the TCP rate as $X[t] = W[t]/d = (C_0R_0 + Q[t])/d$. Recall that in the static channel case, the drop in throughput diminishes as the queue backlog becomes bigger, and Eqn. (6) suggests that there is no incentive to increase transmission power when the backlog exceeds the bandwidth-delay product ($X[t] > 2C_0$). However, in a transition $S_0 \rightarrow S_1$, the increased channel rate in state S_1 may result in $X[t] < C_1$, in which case the drop in throughput would be appreciable. Intuitively, this calls for a *look-ahead* policy which determines power based on current and future events. This effect is captured by the second term in the following throughput evolution equation:

$$\Delta Y = -p_e[t]X[t]\Delta T \max(0, C_0 - X[t]/2) + (1-p_e[t]X[t]\Delta T)p_{01}(\min(X[t], C_1) - C_0)\,. \tag{10}$$

Differentiating with respect to P, we obtain:

$$\frac{\partial \Delta Y}{\partial P[t]} = -f'(P[t])X[t]\Delta T \cdot \max(0, C_0 - \frac{X[t]}{2}) - p_{01}f'(P[t])X[t]\Delta T \cdot (\min(X[t], C_1) - C_0) \tag{11}$$

Eqn. (9) and Eqn. (11) can be combined as

$$\frac{\partial \Delta Y}{\partial P[t]} = -f'(P[t])X[t]\Delta T \max(0, \min(X[t], C_0) - \frac{X[t]}{2}) - f'(P[t])X[t]\Delta T p_{01}(\min(X[t], C_1) - \min(X[t], C_0))\,. \tag{12}$$

Utilizing similar arguments for the alternate transition $S_1 \rightarrow S_0$, we obtain that,

$$\frac{\partial \Delta Y}{\partial P[t]} = -f'(P[t])X[t] \cdot \max(0, C_1 - \frac{X[t]}{2}) - p_{10}f'(P[t])X[t]\Delta T \cdot (C_0 - C_1)\,. \tag{13}$$

Again, both the queuing and non-queuing scenarios can be summarized into a single equation:

$$\frac{\partial \Delta Y}{\partial P[t]} = -f'(P[t])X[t] \cdot \max(0, \min(X[t], C_1) - \frac{X[t]}{2}) - p_{10}f'(P[t])X[t]\Delta T \cdot (\min(X[t], C_0) - \min(X[t], C_1))\,. \tag{14}$$

We can now easily generalize the rate-power relationship to the case when we have an n-state channel. Let i denote the channel state in the slot of interest and p_{ij} the probability of transition to channel j. We can then write:

$$\frac{\partial \Delta Y}{\partial P[t]} = -f'(P[t])X[t]\Delta T \cdot \max(0, \min(X[t], C_i) - \frac{X[t]}{2}) - \sum_j p_{ij}f'(P[t])X[t]\Delta T \cdot (\min(X[t], C_j) - \min(X[t], C_i))\,. \tag{15}$$

The Markovian Channel State Transition based algorithm works in a very similar fashion to our previous proposals, with the exception of utilizing Eqn. (15) in each slot to decide the transmit power as a function of the current channel state $C[t]$ and the TCP state $X[t]$.

C. Threshold Based Algorithm

Both the Moving Average and Markov Channel State Transition algorithms rely on knowledge of the *TCP state* $X[t]$ as well as propagation delay d in order to compute transmit power in each slot. In particular, they require monitoring of both the queue size $Q[t]$ as well as the TCP outstanding window. Although this is practically possible, Eqn. (7) also motivates a very simple alternative that we define as a threshold-based heuristic.

The key insight we derive from Eqn. (1) and Eqn. (5) is that when there is no buffer backlog, TCP is extremely sensitive to channel errors. However, as the backlog grows, TCP sensitivity to channel errors reduces. We incorporate this approach by devising a simple single threshold based approach that works as follows. The heuristic takes as input a pre-defined backlog *threshold* T.

In each slot, assume that the base station makes the *baseline* decision of transmitting at a rate $D[t]$ chosen based on achieving a target FER F at maximum power P_{max}. Denote the effective *achieved* FER with power P_{max} as F_e. Note that the effective achieved FER F_e for a selected rate is usually less than the target FER F, *i.e.,* $F_e \leq F$, because of discrete rate levels.

The Threshold based policy then modifies the baseline policy decisions to chose transmission rate $D'[t]$ and FER F' so as to improve power efficiency in the following fashion:

1) If $Q \leq T$, use the *baseline* decision.
2) If $Q > T$, reduce power as much as possible such that new FER $F' \leq F$.

The explanation for the above actions is straightforward. If the queue length is zero, then as shown in Eqn. (1) TCP is very sensitive to channel errors. Hence, in order to avoid reduction in throughput, the heuristic retains the original decision to transmit at maximum power, since it provides the lowest possible FER (F_e). On the other hand if the queue backlog is greater than T, the policy decides that TCP can tolerate errors, as motivated by our previous analysis. Consequently, it reduces power to match the *target* FER F (as opposed to the effective FER F_e which is usually much smaller).

Hence, by monitoring only the queue length, the policy tries to approximate the desired behavior of a TCP-aware policy. While simplicity is the key feature of this policy, the drawback of such a policy is its agnosticism to channel dynamics. Indeed it does not account for fluctuations in the channel bandwidth which can render backlog-based decisions inaccurate.

Before moving on to evaluation of the algorithms, a note is in order regarding their implementation in practice, especially with respect to time-outs. A common feature across all the algorithms is that of local instantaneous decisions. Specifically, all algorithms monitor the *current state* and then choose power-levels to achieve a certain target. In practice, if the backlog is sufficiently high, this can result in several consecutive packet losses, since the schemes would consider TCP throughput to be insensitive to errors as long as the large backlog persists. While TCP New Reno (the version assumed in this work) reduces rate only once in a round trip even with multiple packet losses, a more problematic aspect is the time-out mechanism which can get triggered quite easily due to several losses from a backlogged buffer. Indeed, in preliminary evaluation of our algorithms we observed this to be the main aspect in performance degradation. To circumvent this we include an additional rule. If the power policy results in a packet drop, we revert to the baseline transmit scheme at maximum power for one round trip time to avoid multiple drops. This approach was found to successfully eliminate spurious time-outs. We next evaluate all three proposed policies in Section V on actual traces collected from a commercial 1xEV-DO network.

V. PERFORMANCE EVALUATION

This section is devoted to exploring the performance of the power-adaptation algorithms proposed in Section IV. We first present a simple fluid equation based numerical model that illustrates the nature of the power-throughput trade-off relationship for TCP and also serves as a fast computational tool for rough estimation purposes. This is followed by a detailed *ns-2* based evaluation of the schemes in several scenarios using a custom 1xEV-DO module that is fed with wireless traces.

A. Numerical Evaluation with a Fluid Model

The main motivation behind developing a simple numerical model is to provide a fast approximate method (as opposed to a detailed event-based simulation) with the dual objective of illustrating the characteristics of the power-throughput trade-off as a function of the parameter γ (see Eqn. (4)) as well as allowing a quick approximation of a desirable range of γ to explore with more exhaustive simulations.

The model is essentially a discrete event, trace driven set of differential equations that takes as input, in each slot, the SINR and channel rate from a wireless trace and then computes TCP state variables based on Eqn. (15), while taking into account channel loss and buffer dynamics. We explain the model in further detail below and then use it to study the system in question.

1) TCP Fluid Model: We assume the TCP version to be New Reno, wherein multiple packet losses in a round-trip time are treated as a single loss event, resulting in only one reduction in the sending rate. The model tracks the evolution of TCP transmission rate $X[t]$ and the queue size $Q[t]$ in each slot t as follows.

Let $L[t]$ denote the probability of having a loss event due to channel errors in time slot t. Then, $L[t]$ is given by

$$L[t] = 1 - (1 - p_e[t])^{X[t]\Delta T} = p_e[t]X[t]\Delta T + o(\Delta T^2),$$

where, $p_e[t]$ is the packet loss probability in time slot t (how to compute $p_e[t]$ is explained later). Consequently, the expected change in TCP sending rate due to transmission losses in time slot t is given by $-\frac{1}{2}p_e[t]X^2[t]\Delta T$.

Let $Q[t]$ denote the queue length at the base station in time slot t. Then the instantaneous round trip time (including queuing delay) denoted by $R[t]$ is given by $R[t] = d + Q[t]/C[t]$, where, d is the propagation delay, and $C[t] = C_i$ if the channel is in state i. We note that in our model, all quantities are expressed in units of packet and second.

With respect to the queue dynamics, three regimes of operation are possible:

(i) $0 < Q[t] < B$

In this case, TCP is coupled to the channel process through transmission losses. The following equations describe TCP and buffer dynamics:

$$\begin{cases}\dot{Q}[t] = X[t] - C[t], \\ \dot{X}[t] = \frac{1}{R[t]^2} - \frac{1}{2}p_e[t]X^2[t]\,. \end{cases}$$

(ii) $Q[t] = 0$
In this case, buffer is empty and remains empty until TCP sending rate becomes greater than the channel rate. The following equations describe TCP and buffer dynamics:

$$\begin{cases}\dot{Q}[t] = (X[t] - C[t])^+\,, \\ \dot{X}[t] = \frac{1}{R[t]^2} - \frac{1}{2}p_e[t]X^2[t]\,. \end{cases}$$

(iii) $Q[t] = B$
In this case, TCP faces a congestion loss due to buffer overflow. Buffer remains full until TCP sending rate drops below the channel rate. Since $Q[t] = B$, it follows that $X[t] > C[t]$. TCP and buffer dynamics are described as follows:

$$\begin{cases}\dot{Q}[t] = (X[t] - C[t])^-\,, \\ \dot{X}[t] = -\frac{1}{2}X[t]\,. \end{cases}$$

In the above equations, we have used the notation $(x)^+ = \max\{x, 0\}$, and $(x)^- = \min\{x, 0\}$. These equations can be solved numerically to compute $X[t]$ and $Q[t]$.

We start at the initial state $(X[0] = 0, Q[0] = 0)$, and provide as inputs, a trace file that lists the observed SINR[4] and channel rate assigned by the system in each slot t as well as a *mapping table* , which for each channel rate, provides the FER versus power curve (*i.e.,* $p_e = f(P)$ relation). The numerical evaluation proceeds as follows.

In each slot t, we record the assigned channel rate $C[t]$. Assuming a specific power-policy and value of the power-throughput trade-off parameter γ, we compute $p_e[t]$ as follows. Utilizing the mapping table for the specific channel rate, we look up the smallest SINR, whose associated FER yields a packet error rate $p_e[t]$ that satisfies Eqn. (4). More specifically for each SINR's associated FER, the packet error probability $p_e[t]$ is computed as:

$$p_e[t] = 1 - (1 - \mathrm{FER}[t])^{\lceil \frac{M}{b[t]} \rceil},$$

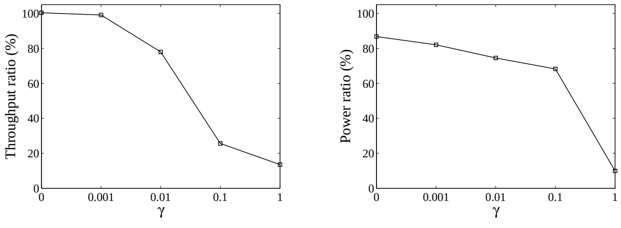
where, M is the TCP packet size and $b[t]$ is the radio frame size. We then plug $p_e[t]$ into the appropriate power-adaptation policy equations (*e.g.,* Eqn. (15)) to determine if it satisfies the threshold. The SINR is continually reduced till the threshold is violated. A loss event is simulated with probability $L[t]$ and the above equation system is utilized to iteratively update $X[t]$ and $Q[t]$.

It is worth mentioning that we have considered more details such as fast recovery and time-outs in our model evaluation but such details are omitted for the sake of clarity.

2) Throughput-Power Trade-off: We now demonstrate how the above model can be utilized to study the system behavior with an example. Let $\overline{X}$ and $\overline{P}$ denote the sample average TCP throughput and average power consumption at the base station. Define *TCP throughput ratio* as follows:

$$\text{TCP throughput ratio} = \frac{\overline{X}_{\text{Adaptive}}}{\overline{X}_{\text{Baseline}}} \times 100\,.$$

The *power consumption ratio* is defined similarly. The *baseline scheme* reflects the power control mechanism implemented in 1xEV-DO networks, wherein the base station transmits at *maximum power* in every time slot, regardless of TCP or the wireless channel condition. For a normalized maximum power of 1 unit, the power consumption of the baseline scheme over T time slots is simply T.



(a) Throughput (b) Power

Fig. 2. Impact of γ on power-throughput trade-off.

Figs. 2(a) and 2(b) show the trade-off between TCP throughput and transmit power with respect to γ with the *Markovian Channel Transition* algorithm (Section IV-B). Further details of the set-up are provided in Section V-B.

It is observed that by intelligently adapting transmission power with respect to TCP dynamics, significant power savings can be achieved without sacrificing TCP throughput. For instance, at $\gamma = 0$, the adaptive scheme results in 15% reduction in transmit power while achieving the same throughput as the baseline.

What this shows is that essentially γ can be used as a control knob to trade off TCP throughput for transmit power. Interestingly, the results obtained from the analytical model match closely with those obtained from *ns-2* simulations presented in next section. However, the model is significantly faster than *ns-2,* and can be used conveniently to choose a proper throughput-power trade-off.

B. Simulation Setup

We next describe details of the setup utilized for a comprehensive study of algorithms utilizing realistic system parameters. For our evaluation, we implement a system similar to the current 1xEV-DO system with the proposed power-control schemes. The system consists of a FTP server, one or more base stations, and one or more mobile nodes attached to each base station, all performing large file downloads from the FTP server through the wireless network. The round trip propagation delay between the server and any mobile node was set to 100 milli-seconds and the buffer size at each base station was set to 40 packets (each packet was 1500 bytes). We consider a single-cell, single-user scenario first followed by a multi-cell, multi-user scenario.

We emulate the 1xEV-DO wireless channel using traces collected from a commercial 1xEV-DO cellular network. Each

---

[4] Recall that the observed SINR is computed using a pilot signal transmitted at *maximum* power P_{max}. Hence, this represents the highest possible SINR.

trace is essentially a time-series that lists the observed SINR computed using the pilot signal. In addition, we also obtained a mapping table that provides the FER versus power curve for each of the 12 channel transmission rates (DRCs) supported by the 1xEV-DO standard.

We consider both single-cell and multi-cell scenarios. In the single-cell scenario, we assume that the neighboring base stations have no power control and always transmit at maximum power. Therefore, in each slot, the interference from neighboring cells remains constant and the amount of power reduction at the reference BS results directly in SINR changes for mobiles in the cell. In the multi-cell scenario, we assume all cells have deployed the same power control scheme.

The system is simulated using *ns-2*. We build an 1xEV-DO module which takes packets from a link's transmission buffer and send them as dictated by the 1xEV-DO scheduler and we implemented the power control schemes in the 1xEV-DO module. Therefore, we perform each simulation run for 110 seconds and ignore the first 10 seconds. In each scenario, we perform the simulation 10 times with different seeds and then calculate the average. The results are presented in subsection V-E for the single cell scenario and in subsection V-F for the multiple-cell scenario.

C. Power Control Schemes

We implement and evaluate the following power control schemes:

The Baseline Scheme ($B(x)$): This simulates current 1xEV-DO systems in which a BS always transmits at full power. In each slot, given a mobile's SINR, the system selects the highest rate that satisfies a pre-specified target FER. For example, in a slot, a mobile can be assigned to any of the following rates with corresponding FERs: 2.4Mbps (FER = 0.5), 1.8Mbps (FER = 0.005), and 1.2 Mbps (FER = 0). If the target FER is 0.01, then the system will assign rate 1.8Mbps to the mobile. We choose four specific target FERs (0, 0.001, 0.01, and 0.1). We call it the Baseline scheme with the target FER x, or $B(x)$. We assume target FER to be 0.01 because it is commonly used in practice. We later compare the power adaption schemes to $B(0.01)$ in terms of throughput ratio and power consumption ratio.

Moving Average Algorithm (MAV): As explained in Section IV-A, MAV approximates the channel capacity based on an exponentially weighted moving average. We set the weighting parameter $\beta = 0.01$.

Instantaneous Channel Rate (INS): In INS, the system evaluates the cost function (Eqn. (7)) utilizing only the *instantaneous* channel capacity and selects the *lowest* power level that satisfies $\frac{\partial \Delta Y}{\partial P} \leq \gamma$. We note that INS is essentially a special case of MAV with $\beta = 1$.

Markovian Channel Transition Algorithm (MCT): This algorithm was presented in Section IV-B and utilizes the Markovian nature of the channel to predict future states. The transition matrix were *a priori* computed for all wireless traces.

Threshold-based Scheme (THR): This scheme is described in Section IV-C and we simulate it with a threshold $T = 5$.

For each power-control scheme, the metrics we are interested in are the throughput as well as power utilized *relative* to the baseline scheme $B(0.01)$.

D. Wireless Channel Characteristics

We evaluated our algorithms on three traces collected on a commercial 1xEV-DO network. All three traces were collected on a laptop attached to a 1xEV-DO data card during a single drive test on a major Interstate Highway traveling around 60 mph, during which the instantaneous SINR computed by the device in each time-slot was logged. The length of each trace was 110 seconds long, translating into about $66,600$ slots (each slot is 1.67 milli-second in 1xEV-DO). Since the traces were collected in a high mobility environment, they exhibit significant variability in channel conditions as well as associated channel rates (computed with a default of 1% FER). Table I presents these metrics for all the three traces. The high channel variability is highlighted in the last column which lists the average time spent in any state. Typically transitions in channel rate occurred every 3 milli-seconds which is very rapid compared to round trip times which are usually of the order of tens of milli-seconds.

TABLE I
CHARACTERISTICS OF WIRELESS TRACES

Trace No.	Avg. SINR (dB)	Avg. Sojourn Time (ms)
1	8.86	3.37
2	7.61	3.6
3	7.7	3.07

E. Single User Evaluation via ns-2

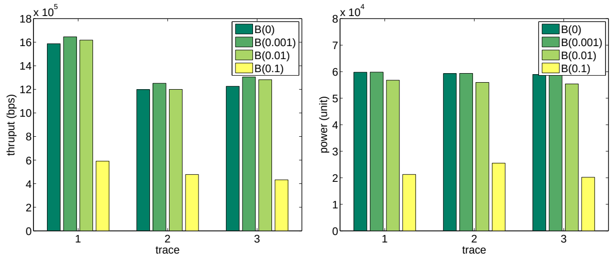
In this subsection, we present simulation results from *ns-2* for the single cell-single user scenario. We first evaluate how a single TCP session performs on a wireless channel without any power control schemes. Table II summarizes the characteristics of the three traces in terms of average channel rate and average *effective* FER given each target FER. We observe that the average effective FER is several magnitudes lower than the target FER.

We use all three traces with four target FER levels. Figure 3 compares TCP throughput and system power consumption in each (trace, target FER) combination. Note that in each slot, the BS transmits to the mobile with full power. Therefore, the power consumption is determined by the number of slots in which the BS is transmitting. BS will not transmit to the MS if and only if 1) there is no data for MS (MS has a small TCP window), or 2) channel rate is zero for MS (MS has a very low SINR). We assume the amount of power consumed in one slot is 1 unit. There are 59980 slots in each simulation. Hence, the maximum amount of power consumption would be 59980 units. We notice that when the target FER is 0, 0.001, or 0.01, both TCP throughput and power consumption are similar. However, with target FER= 0.1, both TCP throughput and power consumption are lower because TCP times out frequently and the channel cannot be fully utilized.

We next simulate the proposed power adaptation schemes. We use $B(0.01)$ (the Baseline scheme with target FER = 0.01)

TABLE II
AVERAGE CHANNEL RATE (MBPS) AND EFFECTIVE FER FOR THREE TRACES, WITH DIFFERENT TARGET FERS.

Trace No.	Average channel rate in Mbps (Average effective FER)			
	0	0.001	0.01	0.1
1	1.665 (0)	1.718 (.00004)	1.836 (.0006)	1.911 (.008)
2	1.175 (0)	1.226 (.00005)	1.296 (.0005)	1.353 (.007)
3	1.202 (0)	1.276 (.00005)	1.367 (.0006)	1.436 (.009)



(a) Throughput (b) Power

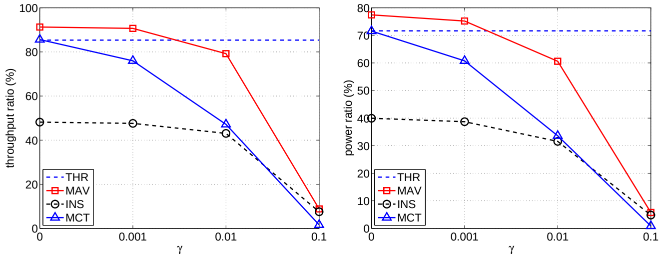
Fig. 3. Single-cell scenario: performance of the baseline scheme with different target FERs.

as the baseline for comparison and compute the throughput and power consumption relative to those of $B(0.01)$. The results are plotted in Figs. 4 and 5 for Trace 1 and Trace 2, respectively, with varying γ values in the power-control schemes. We omit results from Trace 3 because they are similar. From the figures, throughput is maximized at $\gamma = 0$ for the power-control schemes. Hence we focus on the performance at $\gamma = 0$. Among the schemes, the naive scheme (INS) has the poorest performance. It can only achieve 50% of the baseline throughput. By taking into consideration the channel condition in the very next slot, the transition-matrix based scheme (MCT) significantly improves TCP throughput (80% of Baseline) over INS while consuming less or similar power (70% of Baseline). The scheme using a moving average of channel rate (MAV) also performs well (90% of Baseline throughput), however it also consumes more power than MCT ($70-80\%$ of Baseline). Last but not least, the threshold-based scheme (THR), despite its simplicity, achieves similar level of throughput and power consumption to MAV and MCT.

There is no clear winner among MAV, MCT and THR in the single-cell scenario. MAV in general provides higher throughput but also has higher power consumption than MCT or THR. Overall however, across both traces MAV provides more robust performance, while THR is attractive for its simplicity and insensitivity to γ.

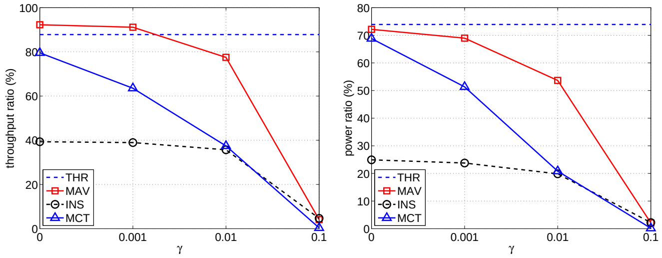
We next look at how each scheme changes the FER, since that is the primary method of power-control. We again focus on $\gamma = 0$. Recall that for $B(0.01)$, the average FER is 0.0006, 0.0005, 0.0006 for three traces, respectively. From Table III, all schemes increases the FER, with INS being the most aggressive one. That also explains why INS performs worst in terms of throughput. On the other hand, MAV produces higher FER than THR and MCT. THR and MCT have similar levels of FER and hence similar throughput performance, while MCT is more power-efficient (Figs. 4 and 5).

We also simulated a multi-user scenario with two users attached to the same base station and scheduled by the *Proportional Fair* algorithm [10]. The results are similar to the single-user scenario and hence omitted.



(a) Throughput (b) Power

Fig. 4. Single-cell scenario: performance of different schemes for Trace 1.



(a) Throughput (b) Power

Fig. 5. Single-cell scenario: performance of different schemes for Trace 2.

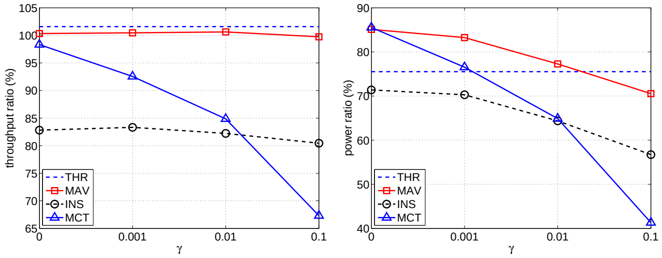
F. Multi-User/Multi-Cell Evaluation via ns-2

We consider three base stations that all have deployed a certain power adaption algorithm and compare the system-wide performance to a baseline case in which all base stations are utilizing the $B(0.01)$ scheme. In each cell, there are N mobile users simultaneously performing long FTP downloads. Each BS schedules the mobiles in its cell according to the *Proportional Fair (PF)* algorithm. In each slot, each BS makes a decision independently of others and chooses the amount of power (P) based on the power-control scheme for transmitting to the mobile selected by the PF algorithm. Then, while transmitting, the (simulation) system recomputes the interference (I) of the scheduled mobiles by averaging the amount of power from neighboring cells (assuming the interference level when all neighboring BS's transmitting at full power is 1 unit) and derives the *actual* SINR ($10\log(P/I)$). This new SINR is used to update the effective FER.

We simulate two cases: $N = 1$ and $N = 2$. When $N = 1$, we assume the user in cell i has the channel condition given by trace i, and $i = 1, 2, 3$. When $N = 2$, we use three additionally

TABLE III
AVERAGE FRAME ERROR RATE (FER) FOR THREE TRACES, PRODUCED BY DIFFERENT POWER CONTROL SCHEMES, AT $\gamma = 0$.

Trace No.	Average FER			
	THR	MAV	INS	MCT
1	0.0013	0.0030	0.0089	0.0016
2	0.0011	0.0034	0.0072	0.0014
3	0.0012	0.0033	0.0082	0.0011



(a) Throughput

(b) Power

Fig. 6. Multi-cell scenario: performance of different schemes.

collected mobile traces (details omitted here) for the other mobile users. We present the performance of the algorithms (relative to the baseline case where all BS's are transmitting at full power) when $N = 2$ in Fig. 6. Due to space limitations, we omit the results from $N = 1$ here since they are similar.

Similar to the single-cell scenario (Section V-E), throughput is maximized at $\gamma = 0$ for the power-control schemes. So we also focus on $\gamma = 0$. We observe from Fig. 6 that both THR and MAV achieve slightly better throughput than the baseline case. However, THR consumes much less power (76%) and is overall better performing. MCT also performs reasonably well by achieving throughput ratio above 98%. MCT and MAV have similar power requirement at around 85%.

To summarize, we observe that the power-aware control schemes MAV, MCT and THR can yield significant power-savings without severely impacting TCP throughput both in single-cell and especially in multiple-cell scenarios with a throughput ratio of $90-100\%$ compared to the Baseline, while utilizing only $70-80\%$ of the Baseline power.

VI. CONCLUSIONS

This paper presents a simple proof-of-concept to show that TCP dynamics can be exploited to improve power efficiency in cellular networks. Specifically, we show that in the presence of large buffers and variable channels, there exist regimes where TCP can tolerate packet errors. With the help of simple analytical models that provide insight into TCP behavior, we developed power adaptation policies that control frame error rates, and thus save power without sacrificing TCP throughput. These policies were evaluated through extensive *ns-2* simulations on real channel traces using practical modulation schemes, and shown to perform quite well in terms of trade-off between transmission power and throughput. Our models did not incorporate feedback delay, or sophisticated and more complete TCP models. A question for the future would be as to how/whether their incorporation would further improve power efficiency.

REFERENCES

[1] QualComm, "1xEV: 1x EVolution, IS-856 TIA/EIA Standard," Available at http://www.qualcomm.com/technology/1xev-do/whitepapers.html.

[2] IEEE, "802.16: WorldWide Interoperability for Microwave Access (WiMax)," Dec. 2001.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM*, Vancouver, Canada, Sep. 1998.

[4] J. P. Singh, Y. Li, N. Bambos, A. Bahai, B. Xu, and G. Zimmerman, "TCP performance dynamics and link-layer adaptation based optimization methods for wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, May 2007.

[5] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," *ACM Wireless Networks*, vol. 8, no. 5, Sep. 2002.

[6] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[7] R. Ludwig and R. H. Katz, "The Eifel algorithm: Making TCP robust against spurious retransmissions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 1, Jan. 2000.

[8] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, no. 4, Dec. 1995.

[9] M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in *Proc. ACM MOBICOM*, Atlanta, USA, Sep. 2002.

[10] ——, "Improving TCP/IP performance over third generation wireless networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004.

[11] C. Barakat, E. Altman, and W. Dabbous, "On TCP performance in a heterogeneous network: A survey," *IEEE Commun. Mag.*, vol. 38, no. 1, Jan. 2000.

[12] H. Elaarag, "Improving TCP performance over mobile networks," *ACM Comput. Surv.*, vol. 34, no. 3, Sep. 2002.

[13] C. Barakat and E. Altman, "Bandwidth tradeoff between TCP and link-level FEC," *Computer Networks*, vol. 39, no. 5, Jun. 2002.

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

[15] 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, Feb. 2006.

[16] 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.

[17] 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, Jul. 2000.

[18] 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, Jan. 2000.

[19] 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, Feb. 2004.

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

[21] 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, Aug. 2000.

[22] 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, Sep. 2006.

[23] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang, "Experience in a 3G network: Interplay betweeen the channel and applications," in *Proc. ACM MOBICOMM*, San Francisco, USA, Sep. 2008.