CDMA Channel Parameters Maximizing TCP Throughput

Francois Baccelli
INRIA-ENS, France

Rene L. Cruz
UCSD, USA

Antonio Nucci
Narus, USA

***Abstract*— We consider a single TCP session traversing a wireless channel, with a constant signal to noise ratio (SINR) at the receiver. We consider the problem of determining the optimal transmission energy per bit, to maximize TCP throughput. Specifically, in the case where direct sequence spread spectrum modulation is used over a fixed bandwidth channel, we find the optimal processing gain m that maximizes TCP throughput. Block codes applied to each packet for forward error correction can also be used, and in that case we consider the joint optimization of the coding rate to maximize TCP throughput. Finally, we discuss the issue of assigning target SINR values. In order to carry out our analysis, we obtain a TCP throughput formula in terms of the packet transmission error probability p and the transmission capacity C, which is of independent interest. In our TCP model, the window size is cut in half for each packet transmission loss, and also cut in half whenever the window size exceeds the transmission capacity C.**

I. Introduction

Cellular wireless networks were originally designed to support voice, which has stringent delay requirements. In these networks, a power control algorithm is used to maintain a target signal to noise ratio (SINR) for each user. The power control algorithm adapts to fast multi-path fading that arises due to mobility of users or the sources of scattering, so that a constant bit rate and a required maximum bit error rate is maintained for each connection, with low transport latency. Thus, for example, when a user encounters a fading channel condition, the transmission power is boosted so that the voice conversation can continue in real time.

These systems have been adapted to carry data as well. A fixed capacity channel may be allocated for a data user. We are generally interested in optimizing the channel parameters in order to provide the best performance for the data user. For simplicity here we assume that a data user corresponds to a single TCP connection. We assume that the channel is allocated a SINR which will be maintained to a constant target value to be determined in function of the network conditions. We focus on the case of a long lived TCP connection. The target SINR value may be adapted over time to respond to user mobility, and we consider the regime where the TCP connection reaches a steady state between updates of the target SINR value.

We consider the case of code-division multiple access (CDMA), and address the problem of optimizing the processing gain m and coding rate ρ to maximize TCP throughput. By adjusting the processing gain and coding rate, we trade-off the bit transmission rate C and packet transmission error rate p. By making the processing gain m large we increase the energy per bit and so the packet transmission loss probability p is small. However, the bit transmission rate C is proportional to $1/m$.

To understand the nature of the optimization we consider, it is useful to think of two extremes. At one extreme we can make the bit-transmission rate C high, but the packet transmission error rate p will be large. Packet transmission errors will generally cause the TCP protocol to reduce its window size, and in turn decrease throughput. At the other extreme we can make the packet transmission error rate p very small, at the expense of reduced transmission rate C. The bit transmission rate C ultimately limits the TCP window size, since buffer overflows will occur when the window gets sufficiently large, and TCP will cut the window size in half, responding to the congestion that occurs at the buffer. Thus, TCP throughput is small at this extreme as well.

In order to find the optimal operating point, we consider a model for analyzing the TCP throughput where packet losses due to transmission errors and packet losses due to congestion events are distinguished. We consider a fluid model where the window size is cut in half for each packet loss due to a transmission error or buffer overflow event. The packet loss probability is p, and we assume that a congestion event happens when the window size reaches a value that matches the bit-transmission rate, C, on the channel. This model is appropriate for small buffers. This makes sense within e.g. the CDMA context where the bit-transmission rate is small, compared to what is available on wireline networks, and where large downlink buffers would imply large RTTs and hence poor TCP performance as TCP throughput is known to be roughly inversely proportional to RTT.

For this model, we find a formula for the TCP throughput as a function of p and C, which is of independent interest. This is a generalization of the well known“square root” formula for TCP throughput where packet transmission errors and buffer overflows are not separately modeled. We also obtain formulas for the probability density function of the TCP window size.

We find that TCP throughput can be fairly sensitive to the physical layer parameters, i.e. the processing gain m or symbol alphabet size M. This suggests it may be advantageous to adapt these parameters according to the environment. In particular, they can be adapted according to the target SINR and the round-trip time (RTT) in order to maximize TCP throughput.

In the last section, we also discuss the problem of assignment of a target SINR values, and specifically discuss the case of a wireless cellular CDMA downlink.

Before describing our model and analyzing it in more detail, we first discuss some related work. Many authors have considered the general problem of performance of the TCP protocol over wireless links. A comparison of various approaches to the problem is given in [4]. One branch of related work is concerned with the problem of determining whether packet losses are due to congestion or due to transmission errors, so that TCP can go into congestion avoidance mode only when packet losses are due to congestion. A second branch of work considers the approach of splitting the TCP connection at the wire-line/wireless infrastructure boundary, so that the TCP connection is isolated from the packet transmission errors on the wireless channel. A third approach to optimizing TCP performance over wireless channels is to optimize the link layer for TCP performance. The approach we take in this paper falls into the third category. Within this framework, Liu, Goeckel, and Towsley [7] considered the problem of adapting the coding rate ρ to the channel conditions, with the objective of maximizing TCP throughput. It was already noted in [7] that optimal values of operating parameters for the channel for TCP were different than those for UDP. The main novelty of the present paper is the fact that it provides an analytic framework for this adaptation.

II. TCP Throughput

A. Model

In this section, we analyze the throughput of a single TCP flow over a wireless channel. Each packet contains L bits, and when it is transmitted over the channel, it is lost with probability p. We assume that packet transmission errors are independent. This is reasonable, since we assume that the channel has a fixed signal to interference plus noise ratio γ. Later on, in Section IV, we will consider how the packet transmission loss probability p depends on γ and other physical layer parameters. We assume it takes L/C seconds to transmit each packet on the channel. The parameter C is called the raw transmission capacity and has units of bits/sec.

We now consider a fluid model for the TCP flow. Our model is in terms of the parameters C and p discussed above. Consider a random process $X(t)$ that models the instantaneous throughput for the TCP flow. The instantaneous throughput is assumed to be proportional to the TCP window size. Our model for the dynamics of $X(t)$ is as follows. Let R be the round trip delay (assumed constant here), in units of seconds. When there is no loss, at time t, $X(t)$ increases at rate L/R^2. If $X(t)$ reaches C, a congestion event occurs, causing $X(t)$ to be reduced in half to the value $C/2$. Packet losses are modeled by a time in-homogeneous Poisson process, where the rate of loss at time t is $\lambda(t) = pX(t)/L$. This reflects the fact that the rate of packet loss is higher when the packet rate is higher. When there is a packet loss at time t, the value of $X(t)$ is reduced in half.

Let $f(x)$ be the stationary probability density function for the instantaneous throughput $X(t)$ of the TCP connection. It is shown in [2] that the density f satisfies the differential equation

$$\frac{df(x)}{dx} = \begin{cases} -\alpha x f(x) & \text{if } C/2 < x \le C \\ -\alpha x f(x) + 4\alpha x f(2x) & \text{if } 0 \le x < C/2 \end{cases}, \tag{1}$$

where $\alpha = pR^2/L^2$. The density $f(x)$ is discontinuous at $x = C/2$ and such that

$$f((C/2)^+) - f((C/2)^-) = f(C^-) \; . \tag{2}$$

B. Mean Values

In the following theorem we obtain a closed form equation for $TCP(C,p)$, which leads to a simple approximation for $TCP(C,p)$. Let $\widehat{f}(u)$ be the Mellin transform of $f(x)$:

$$\widehat{f}(u) \;=\; \int_0^C f(x) x^{u-1} dx \tag{3}$$

with $u \ge 1$. Let $\Gamma(u)$ be the Mellin transform of e^{-x}, i.e.

$$\Gamma(u) = \int_0^\infty e^{-x} x^{u-1} dx \; .$$

For all $l \ge 0$, define

$$\Pi_l(u) = \prod_{k=0}^{l} \left(1 - 2^{-u-2k}\right). \tag{4}$$

The following is proved in [2].

Theorem 1 (Mean TCP Throughput): The Mellin transform of the probability density $f(x)$ of the instantaneous TCP throughput is given by

$$\widehat{f}(u) = \frac{\sum_{l\ge 0} \Pi_l(u) C^u \left(\frac{\alpha C^2}{2}\right)^l \frac{\Gamma(u/2)}{\Gamma(u/2+l+1)}}{\sum_{l\ge 0} \Pi_l(1) C \left(\frac{\alpha C^2}{2}\right)^l \frac{\Gamma(1/2)}{\Gamma(1/2+l+1)}}. \tag{5}$$

In particular, the mean TCP throughput is

$$TCP(C,p) = \frac{\sum_{l\ge 0} \Pi_l(2) C^2 \left(\frac{\alpha C^2}{2}\right)^l \frac{1}{(l+1)!}}{\sum_{l\ge 0} \Pi_l(1) C \left(\frac{\alpha C^2}{2}\right)^l \frac{\sqrt{\pi}}{\Gamma(1/2+l+1)}} \; . \tag{6}$$

From (6) we obtain the following approximation:

Corollary 2 (Approximation to Mean TCP Throughout): The mean TCP throughput satisfies

$$\begin{aligned} TCP(C,p) \;&=\; \frac{3C}{4} - \frac{pR^2C^3}{L^2}\frac{11}{256} - \frac{p^2R^4C^5}{L^4}\frac{497}{491520} \\ &\quad + \frac{925667}{377487360}\frac{p^3R^6C^7}{L^6} + o(p^3R^6C^7/L^6)C. \end{aligned} \tag{7}$$

As expected, the values of the throughput are insensitive to the value of C if the packet loss probability p is sufficiently high. In fact we have the following corollary:

Corollary 3: For all $p > 0$ we have

$$\lim_{C\to\infty} TCP(C,p) = \sqrt{\frac{2}{\alpha}}\frac{1}{\sqrt{\pi}}\frac{\Pi_\infty(2)}{\Pi_\infty(1)} = \frac{1.309}{\sqrt{p}R/L}, \tag{8}$$

The result in Corollary 3 coincides with [3], which is expected since letting C approach ∞ is equivalent to considering random transmission losses only.

III. Optimization of Wireless Channel Parameters

In this section we consider optimization of the wireless channel parameters to maximize the throughput of the TCP session that passes through it. We can use the formula for the TCP throughput in (7) to evaluate $TCP(C, p)$ as a function of C and p. The wireless channel parameters will determine C and p, and we wish to maximize $TCP(C, p)$ with respect to these parameters.

In the context of CDMA systems, a processing gain m is used to adjust the transmission data rate, or equivalently the transmission energy used per bit. A large processing gain is generally needed to compensate for low SINR. We shall consider the problem of optimizing the processing gain to maximize TCP throughput. We shall also consider the use of error correction codes, and jointly optimize the coding rate and the processing gain.

In the next subsection, we present a wireless channel model, which yields formulas for p and C as a function of the wireless channel parameters .

A. Wireless Channel Model

We first consider a model appropriate for a low SINR at the receiver, in the context of a CDMA network.

1) Code Division Multiplexing without FEC: We will consider the case of code-division multiplexing with direct sequence spread spectrum modulation. In this case, we assume that signals are sent with direct sequence spread spectrum modulation, with a chip duration of T_c seconds. Binary Phase Shift Keying (BPSK) is assumed as the underlying modulation scheme. Each bit transmitted is encoded into m chips using a spreading sequence, where m is called the *processing gain*. The bit transmission rate is thus

$$C = \frac{1}{mT_c}. \quad (9)$$

Typically, many users may transmit at the same time, causing interference at each users receiver. Each user has a receiver which correlates the incoming signal with the spreading sequence used at the transmitter. Let γ be the signal to noise plus interference ratio (SINR) at the output of the correlator. We shall define γ mathematically when we discuss assignment of SINR values later.

If we model the interference at the correlator output as Gaussian, the probability of a bit error is

$$BER = Q(\sqrt{m\gamma}), \quad (10)$$

where $Q(\cdot)$ is the CDF of the zero mean unit variance Gaussian density, i.e. $Q(x) = (1/\sqrt{2\pi}) \int_x^\infty e^{-x^2/2} dx$.

In the uncoded case that we consider here (no FEC), each packet corresponds to transmission of L bits on the channel. The probability of a packet error is

$$p = 1 - (1 - Q(\sqrt{m\gamma}))^L. \quad (11)$$

We will use the following approximation for p, which is valid when BER is small:

$$p \approx LQ(\sqrt{m\gamma}) \ . \quad (12)$$

2) Code Division Multiplexing with FEC: In the case where a block code for forward error correction (FEC) is used, the L bits of each packet are encoded into N bits, where $N \geq L$. The ratio $\rho = L/N$ is called the *coding rate*. Since we count throughput in terms of information conveyed, the bit transmission capacity in this case is

$$C = \frac{\rho}{mT_c} \ . \quad (13)$$

The collection of N bits is called a codeword, and there is a codeword for each of the 2^L possible bit patterns of a packet. In general, the N codeword bits contain redundancy, so that if bit transmission errors occur, the bit pattern of the originally encoded packet can sometimes be recovered at the receiver. We assume that up to t errors can be corrected, and that $t+1$ or more bit transmission errors result in a packet loss. The probability of a packet error is thus

$$p = \sum_{j=t+1}^{N} \binom{N}{j} [Q(\sqrt{m\gamma})]^j [1 - Q(\sqrt{m\gamma})]^{N-j} \ .$$

Define $\theta = (t+1)/N$. We use the following large deviations-based estimate for p:

$$p \approx 2^{N[h(\theta) + \theta \log_2(q) + (1-\theta)\log_2(1-q)]} \ , \quad (14)$$

where $q = Q(\sqrt{m\gamma})$ and $h(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the binary entropy function. Notice that this approximation uses the dominant term of Stirling's formula, and that better expansions could be used.

In order to determine t, or equivalently θ, we use the bound [5], which implies that for a given coding rate ρ, there exists a block code with "error correction capability" θ, where θ satisfies

$$\rho = 1 - h(2\theta) \ . \quad (15)$$

B. Optimization of TCP Throughput

In this subsection we consider optimization of TCP throughput in the two cases outlined in the previous subsection. First we consider a single TCP session i which is assigned a given SINR value $\gamma_i = \gamma$. We are interested in optimizing the processing gain $m_i = m$ and the coding rate $\rho_i = \rho$ in order to maximize the mean TCP throughput. Recall that we have $C = \frac{\rho}{mT_c}$, and that the packet error probability p is a decreasing function of m and an increasing function of ρ. We wish to maximize $M = TCP(C, p)$ over all possible values of m and ρ.

1) Code Division Multiplexing without FEC: We first present some numerical results. In Figure 1, we consider the case where $\gamma_i = 0.03$ and the RTT value is $R = 0.1$ second. On the top right curve we plot the TCP throughput as a function of the processing gain m. We see that there is fair amount of sensitivity to the processing gain, and there is a unique maximum around $m = 450$. To get a better sense of this, on the top left plot we illustrate the TCP throughput ISO curves, i.e. values of C and p that yield the same TCP throughput. Superimposed with these ISO curves, we plot the locus of (p, C) values corresponding to different values of processing gain. We see that the optimum processing gain corresponds to crossing a "knee" of a TCP ISO curve. For reference, we consider optimization of the "UDP throughput" $C(1-p)$ in the bottom two plots in Figure 1.

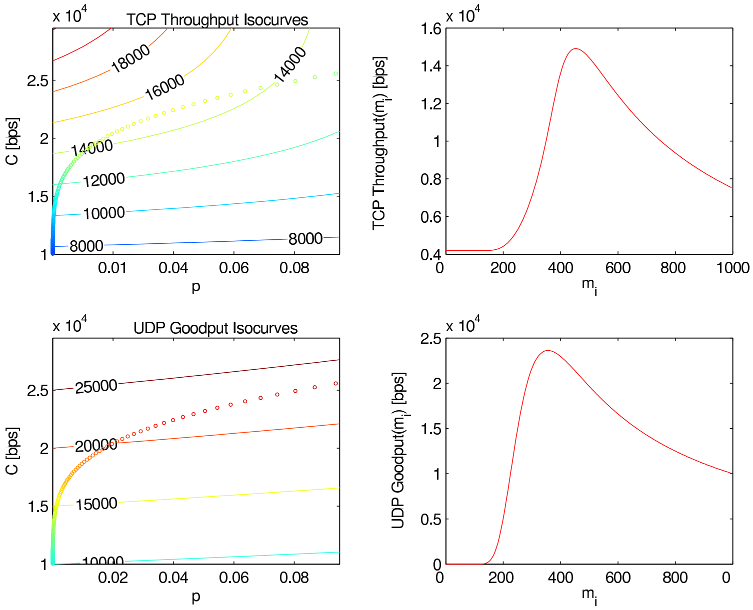

Fig. 1. Example: Cross-Layer Optimization Results - Uncoded Case. $\gamma_i = 0.03$, $R = 0.1$ second

In this case we use (9) for C and (12) for p in the expansion for $TCP(C,p)$ in (7), for an explicit optimization of the mean throughput.

Using the approximation $Q(x) \approx \frac{1}{\sqrt{2\pi x}} e^{-x^2/2}$ in (12), we have

$$p \sim L e^{-\frac{m\gamma}{2}} \frac{1}{\sqrt{2\pi m\gamma}}. \tag{16}$$

If $\alpha C^2/2$ is small, then the first order expansion corresponding to the first two terms in (7), gives the following expression for the mean rate

$$TCP(C,p) \sim \frac{1}{4T_c}\left(\frac{3}{m} - \frac{1}{T_c^2}\frac{\beta}{m^3\sqrt{m}} e^{-\frac{m\gamma}{2}}\right), \tag{17}$$

with $\beta = \frac{11R^2}{64L\sqrt{2\pi\gamma}}$. Differentiating with respect to m, we get that the optimal m solves

$$\frac{1}{4T_c}\frac{1}{m^2}\left(-3 + \frac{\beta}{2T_c^2}\left(\frac{\gamma}{m\sqrt{m}} + \frac{7}{m^2\sqrt{m}}\right)e^{-\frac{m\gamma}{2}}\right) = 0,$$

that is

$$\boxed{e^{-\frac{m\gamma}{2}} = \frac{6T_c^2}{\beta}\frac{m\sqrt{m}}{\gamma + \frac{7}{m}}.} \tag{18}$$

Taking $L = 320$, $R = .1$, $\gamma = 3\ 10^{-2}$, $T_c = 10^{-7}$, we get $\beta \sim 1.15\ 10^{-6}$; so using the first order approximation for the TCP throughput, the optimal processing gain m satisfies

$$e^{-.015m} = 4.727\ 10^{-9}\frac{m\sqrt{m}}{.03 + \frac{7}{m}}. \tag{19}$$

The solution is $m = 459$. The associated value for $\alpha C^2/2$ is $pC^2R^2/(2L^2) \sim 0.815$ which, upon examination of (7), justifies the use of the few loss approximation (the correction brought by the second order term is appr. 4/1000 of the value given by the first order approximation).

Remark 1: Our model has an interesting relation to [6], where all packet transmission errors are hidden with a link layer mechanism (eg link layer ARQ), so that the only packet losses that TCP reacts to are buffer overflows at the link layer buffer.

Let W_{bd} be the bandwidth delay product (where bandwidth is defined in units of TCP packets/sec). The authors of [6] propose sizing the buffer so that the overflow probability q satisfies $q(W_{bd})^2 = 1$. With our method, we find the optimal operating point at $\alpha C^2/2 = 0.815$, which is equivalent to $p(W_{bd})^2/2 = 0.815$.

2) Code Division Multiplexing with FEC: In this case, we wish to jointly optimize the processing gain m and coding rate ρ in order to maximize TCP throughput. In this case we use (13) for C and (14) for p, where θ is determined from ρ using (15).

In Figure 2, we again consider the case where $\gamma = 0.03$ and the RTT value is $R = 0.1$ second, and plot results for when *both* the processing gain m and the coding rate $\rho = L/N$ are jointly optimized. The top left plot shows how the optimum processing gain m^* changes as a function of the coding rate ρ. On the graph we have also labeled the approximate value of the TCP throughput corresponding to each point. For example, for a coding rate of about 0.4, the optimum processing gain is about 100, and the corresponding TCP throughput is about 31Kbps. For comparison, in the top right plot we show how the optimum processing gain m^* changes as a function of the coding rate ρ, where we optimize UDP goodput, i.e. $(1-p)C$. In the bottom left plot we show the optimal TCP and UDP throughput as a function of the coding rate ρ. In the bottom right plot we show the optimum TCP and UDP throughput as a function of the processing gain m.

Next, we describe a procedure for an explicit joint optimization. Define $J = TCP(C,p)$. We are interested in maximizing J with respect to the processing gain m and the coding rate ρ. Note that there is a one-to-one relation between the coding rate ρ and the parameter θ, through equation (15). For convenience we optimize the TCP throughput J with respect to m and θ, and then the optimal coding rate ρ^* is determined by θ^* through (15). Note that θ^* and m^* satisfy $\frac{\partial J}{\partial m} = 0$ and $\frac{\partial J}{\partial \theta} = 0$. Define $F_1(m,\theta) = \frac{\partial J}{\partial \theta}$ and $F_2(m,\theta) = \frac{\partial J}{\partial m}$. In [2], we provide the calculation of F_1 and F_2. We can solve the two non-linear equations $F_1(m,\theta) = 0$ and $F_2(m,\theta) = 0$ in order to determine the optimal processing gain m^* and the optimal value of the coding rate ρ^*, where ρ^* is found from θ^* by using equation (15).

IV. Assignment of SINR Values

In this section, we remind the scenario of the downlink of a cellular CDMA network and discuss issues relating to the assignment of SINR values to the different users.

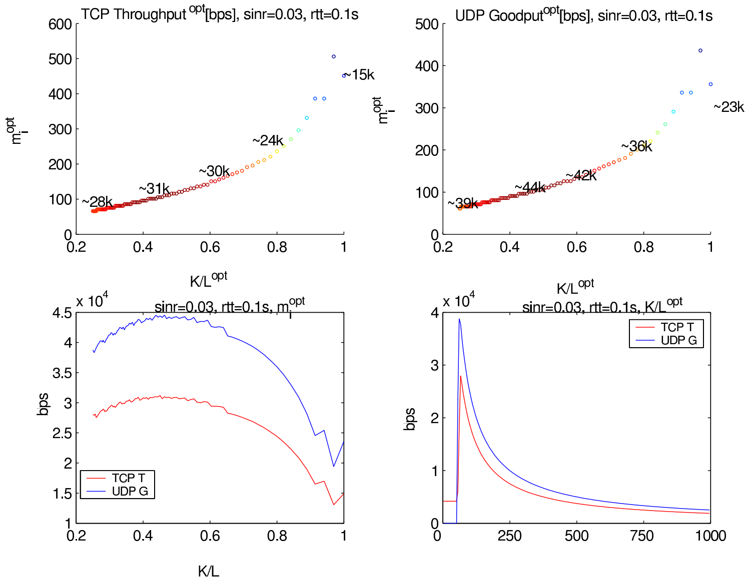

Fig. 2. Example: Joint Optimization of Processing Gain m_i and Coding Rate K/L. $\gamma_i = 0.03$, $R = 0.1$ second

Suppose there are multiple users on the downlink of a CDMA cellular system, where the users may be associated with different base stations. The signal transmitted for the i^{th} user at the associated base station is denoted by $s_i(t)$ is given by

$$s_i(t) = \sqrt{P_i} p_i(t) b_i(t)$$

where $p_i(t)$ is the spreading code, $b_i(t)$ is the data signal, and P_i is a constant. The spreading code $p_i(t)$ takes on values in $\{-1, 1\}$ and is constant over intervals of duration T_c. Specifically,

$$p_i(t) = \sum_{k=-\infty}^{\infty} c_k^i u(t/T_c - k),$$

where $u(x) = 1$ if $0 < x < 1$ and $u(x) = 0$ otherwise, and for each user i, the elements of the sequence $\{c_k^i\}_{k=-\infty}^{\infty}$ are either $+1$ of -1. The constant T_c is the chip duration and is constant across users.

The data signal $b_i(t)$ is also taking its values in $\{-1, 1\}$. For the i^{th} user, we assume that each bit to be sent is repeated m_i chips, where m_i is an integer. Specifically, we have

$$b_i(t) = \sum_{n=-\infty}^{\infty} b_n^i u(t/(m_i T_c) - n)$$

The quantity m_i is called the processing gain for user i. The data rate for user i is thus $1/(m_i T_c)$. Note that $(s_i(t))^2 \equiv P_i$, and so the parameter P_i is called transmission power for the i^{th} user.

Typically either pseudo-random or known deterministic sequences are used to define the spreading codes. For purposes of analysis here we assume that $\mathbb{P}\{c_k^i = 1\} = \mathbb{P}\{c_k^i = -1\} = 1/2$ for all i, k, and that c_k^i is independent of $c_{k'}^{i'}$ if $i \neq i'$ or $k \neq k'$. We shall also assume that the data bits are random and independent, i.e. $\mathbb{P}\{b_n^i = 1\} = \mathbb{P}\{b_n^i = -1\} = 1/2$ for all i, n, and that b_n^i is independent of $b_{n'}^{i'}$ if $i \neq i'$ or $n \neq n'$.

We assume a so-called flat fading (i.e. frequency non-selective) channel model. Let $\sqrt{g_{ki}}$ be the signal path gain from the base station associated with user k to the location of user i. For example, the useful signal at user i is given by $\sqrt{g_{ii}} s_i(t)$. However, the signal intended for another user k, namely $\sqrt{g_{ki}} s_k(t)$ also arrives at the location of user i, with possibly a time shift reflecting the different distances between base stations and users. In addition, an external white Gaussian noise signal $n_i(t)$ is also present at the receiver for user i, with two-sided power spectral density N_0^i. The total signal at the receiver of user i is

$$r_i(t) = \sqrt{g_{ii}} s_i(t) + \sum_{k:k\neq i} \sqrt{g_{ki}} s_k(t - \alpha_k^i) + n_i(t).$$

The numbers α_k^i characterize the propagation delays between transmitters and receivers. Approximating the interference terms as Gaussian, a standard analysis yields that the probability of a bit error, BER, is

$$BER = Q\left(\sqrt{m_i \Gamma_i}\right) , \quad (20)$$

where

$$\Gamma_i = \frac{g_{ii} P_i}{(2/3)(\sum_{k:k\neq i} g_{ki} P_k) + (N_0^i / T_c)} . \quad (21)$$

The quantity Γ_i is called the signal to interference plus noise ratio, SINR. The quantity $m_i \Gamma_i$ is known as the "Eb/No", or energy per bit per noise power density.

Due to mobility of users, the gain values $\{g_{ki}\}$ change with time. We assume that a closed loop power control algorithm is used, to vary the power values $\{P_i\}$ to maintain the SINR values $\{\Gamma_i\}$ at prescribed values.

Define the SINR vector $\vec{\Gamma} = [\Gamma_1, \Gamma_2, \ldots, \Gamma_N]^T$. We say that an SINR vector $\vec{\gamma}$ is *feasible* if there exists a set of non-negative power values $\{P_i\}$ such that $\vec{\Gamma} = \vec{\gamma}$. If $\vec{\gamma}$ is feasible, then the power control algorithm sets the transmission powers $\{P_i\}$ accordingly to achieve the target SINR vector $\vec{\gamma}$.

Next we examine the feasibility condition for a target SINR vector. If $\vec{\Gamma} \geq \vec{\gamma}$ then for all i we have

$$\frac{g_{ii} P_i}{(2/3)(\sum_{k:k\neq i} g_{ki} P_k) + \sigma_i^2} \geq \gamma_i ,$$

where $\sigma_i^2 = N_0^i / T_c$. Equivalently, we have

$$P_i - (2/3)\gamma_i \sum_{k:k\neq i} (g_{ki}/g_{ii}) P_k \geq \gamma_i \sigma_i^2 / g_{ii} ,$$

or in matrix notation, $\vec{P} - F\vec{P} \geq \vec{b}$, where $F = \{F_{i,j}\}$ is an $N \times N$ matrix with $F_{ii} = 0$ and $F_{i,j} = (2/3)\gamma_i (g_{ji}/g_{ii})$ if $i \neq j$, and

$$\vec{b} = [\gamma_1 \sigma_1^2 / g_{11}, \gamma_2 \sigma_2^2 / g_{22}, \ldots, \gamma_N \sigma_N^2 / g_{NN}]^T .$$

There exists a non-negative finite $\vec{P}$ satisfying the above, if and only if the spectral radius of the matrix F is less than unity. In this case, the minimal $\vec{P}$ satisfying the above is

$\vec{P}^* = (I-F)^{-1}\vec{b}$. The minimal power vector $\vec{P}^*$ can be found by an iterative distributed algorithm. There may be additional constraints on the power vector, e.g. there is generally a peak power constraint for each base station.

A simple sufficient condition for the spectral radius of F to be less than η is that each row sum of F is less than η. In general, we can set η strictly less than 1 as a safety margin, as suggested in [1]. Setting $\eta = 1$, $\vec{\gamma}$ is feasible if for all i we have

$$(2/3)\gamma_i \sum_{k:k\neq i} (g_{ki}/g_{ii}) \leq 1 \ . \tag{22}$$

Note that $g_{ki}/g_{ii} = 1$ for all users k that are associated with the same base station that user i is. Furthermore, the value of g_{ki}/g_{ii} is the same for all users k which are associated with the same base station. Suppose there are B base stations. Let b_i be the base station associated with user i. Assuming $b \neq b_i$, define $\alpha_b(i) = g_{ki}/g_{ii}$, where k is such that user k is associated with base station b. Let N_b be the number of users associated with base station b. Defining $N_i^{int} = \sum_{k:k\neq i}(g_{ki}/g_{ii})$, we thus have

$$N_i^{int} = N_{b_i} - 1 + \sum_{b:b\neq b_i} \alpha_b(i) N_b.$$

We call N_i^{int} the effective number of interfering users for user i. The attenuation factor $\alpha_b(i)$ can be measured by user i by comparing the power received in pilot tones from base station b and it's assigned base station. The values of N_b can be considered to be slowly varying, and reported directly to the base station associated with a given user i. If each user i reports the measured value of $\alpha_b(i)$ to its associated base station for all b, then the value of N_i^{int} is known to the base station associated with user i, b_i. Hence base station b_i can calculate an appropriate value for the target SINR γ_i. In particular, we can set $\gamma_i = \frac{1.5\eta}{N_i^{int}}$, where η is a parameter set to a number strictly less than unity, as a safety factor. Note that in general, since the matrix F varies with time, the feasibility of a set of target SINR values also changes with time.

In summary, we a set of target SINR values for the users is specified by the vector $\vec{\gamma}$. A target vector $\vec{\gamma}$ is feasible if and only if the spectral radius of F is less than one. Alternatively, we can use (22) as the basis for allocating target SINR values γ_i to the users.

V. Conclusion

In this paper we have considered the optimization of wireless channel parameters in order to optimize the throughput of a single TCP connection passing through the channel. We assumed that the signal to noise ratio at the receiver of the channel is held constant. We found that the TCP throughput can be fairly sensitive to these parameters, and our results suggest that these parameters should be set carefully and as a function of the system scenario. For example, when the round trip time (RTT) is large, the impact of a packet loss is higher, so the energy per bit should generally increase with increasing RTT. We have also found that the use of forward error correction, when the coding rate is optimized, can significantly increase TCP throughput.

In a network context, the assignment of SINR values can be a very complex problem, particularly when the path gains are changing quickly with time. The results of this paper can perhaps be used as a foundation for considering various approaches for assignment of SINR values, where the interaction between users is explicitly taken into account.

Acknowledgments

We would like to thank Sprint Laboratories, where this paper was initiated, and particularly J. Bolot for his inputs on the questions discussed here.

References

[1] F. Baccelli, B. Blaszczyszyn and M. Karray, "Up and Downlink Admission and Congestion Control and Maximal Load in Large Homogeneous CDMA Networks" *ACM MONET*, vol. 9, no. 6, Dec. 2004.

[2] F. Baccelli, R. Cruz and A. Nucci, http://www.inria.fr/rrrt/rr-5652.html

[3] F. Baccelli, D. R. McDonald, and J. Reynier, "A Mean-Field Model for Multiple TCP Connections through a Buffer Implementing RED," *Performance Evaluation*, vol. 49, pp. 77–97, 2002.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz. "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. on Networking*, 5(6):756-769, 1997

[5] R.E. Blahut, *Theory and Practice of Error-Control Codes*, Addison-Wesley, Reading, Massachusetts, 1983.

[6] H. Chaskar, T. V. Lakshman and U. Madhow, "TCP Over Wireless with Link Level Error Control: Analysis and Design Methodology", *IEEE/ACM Trans. Networking*, vol. 7, no. 5, pp. 605-615, October 1999 http://www.ece.ucsb.edu/Faculty/Madhow/publications.html

[7] B. Liu, D. L. Goeckel and D. Towsley, "TCP-cognizant adaptive forward error correction in wireless networks," *GLOBECOM 2002 - IEEE Global Telecommunications Conference*, vol. 21, no. 1, November 2002 pp. 2139-2144 http://www.cs.uml.edu/ bliu/pub/wirelessTCP.pdf