Wireless Link Adaptation Policies: QoS for Deadline Constrained Traffic with Imperfect Channel Estimates

Tim Holliday, Andrea Goldsmith, and Peter Glynn, Stanford University
thollida@stanford.edu, andrea@systems.stanford.edu, glynn@stanford.edu

Abstract—We present an optimal power and rate control policy for delay constrained traffic in next generation TDMA wireless systems. Our solution minimizes average transmit power while satisfying a constraint on the distribution of packets lost to deadline expiration. We also provide a means to account for erroneous and delayed channel estimates. Our results show the optimal power and rate adaption may change dramatically as mobile speed and channel estimate delay increase. Finally, we present results from a simulation of a GSM EDGE mobile. This simulation incorporates industry standard wireless channels and performance data available from the Third Generation Partnership Project. When compared to the standard Fixed-SIR power control policy, our algorithm provides a significant reduction in power consumption and mitigates some of the negative effects of delayed channel estimates.

I. Introduction

The introduction of third generation (3G) wireless systems provides the opportunity to develop a wide range of new mobile data services. Many of the proposed services (e.g. Voice over IP, streaming audio, and other multimedia) include data that require guarantees on throughput, delay, and probability of data loss. However, wireless links are inherently unreliable since the channel is subject to time-varying interference, fading, and other effects. In order to combat adverse link conditions mobile radios can adapt system parameters (e.g. transmitter power, code rate, modulation, etc.) to the wireless channel and quality of service (QoS) requirements.

In this paper we present several contributions to the field of adaptive power and rate control. First, we present a general dynamic programming framework for the creation of optimal power and rate control policies that satisfy deadline based QoS constraints. Second, we develop within this framework a means to explicitly account for delayed and inaccurate channel estimates. Many link adaptation solutions do not account for erroneous channel estimates. Rather, the link layer control is chosen assuming the channel estimate is correct, and the impact on system performance is determined via calculation or simulation [9]. In our formulation, effective use of information about the wireless channel can greatly improve the value of erroneous estimates. Third, we show significant performance gains over traditional fixed rate algorithms in both power consumption and the tolerable amount of channel estimation error.

In recent years a large body of research has focused on QoS constrained link adaptation algorithms. In [11] the authors formulate a stochastic optimization problem for throughput and backlog sensitive power management. Several authors [1,4,6] have also proposed dynamic programming solutions. In these formulations a reward or penalty function is created for the successful transmission of data at different power levels. This function is then optimized over a finite horizon and the performance of the resulting policies is evaluated via simulation.

As noted in [11], throughput constrained QoS algorithms can add significant delay to data in the presence of slow fading channels, bursty traffic, or inaccurate channel estimation – rendering the algorithm ineffective for delay sensitive data. For voice and multimedia, many delay constrained QoS algorithms attempt to guarantee a Signal to Interference Ratio (SIR) or fixed bit rate over the wireless channel, even though multimedia traffic is not necessarily a fixed rate source. While this type of fixed rate guarantee provides the required QoS, it does not provide the best performance [1] in terms of power consumption. Since mobile wireless systems are generally interference limited, this also implies fixed SIR algorithms may use more system capacity than necessary.

This paper presents an example of an uplink link adaptation algorithm for delay sensitive traffic over TDMA systems. We assume that the network level QoS constraint translates to a deadline requirement for each packet. If a packet is not successfully transmitted by its deadline, it is dropped from the transmitter's buffer. Our link adaptation algorithm produces a power, coding, modulation, and scheduling policy via a dynamic programming recursion adapted to the wireless channel and traffic characteristics. The goal of the dynamic program is to minimize transmitter power while providing a guarantee on the probability of dropping a packet or consecutive packets.

The rest of this paper is organized as follows. First, we present the basic problem formulation and system model for a TDMA uplink channel. Second, we develop an infinite horizon dynamic programming algorithm for computing optimal link adaptation policies. Third, we present empirical results that examine the impact of channel characteristics and estimate delay on control policies and power consumption. Finally, we discuss future research directions and extensions of this model.

II. Problem Formulation and System Model

The mathematical notation in this paper will adhere to the following convention: scalar variables will be Latin characters (e.g. s), random variables will be bold lower case Latin characters (e.g. $\mathbf{x}$), sets will be bold upper case Latin letters (e.g. $\mathbf{A}$), and vector quantities will have a right arrow superscript over the letter (e.g. $\vec{a}$).

Consider a single wireless mobile transmitting deadline sensitive data to a base station. The base station is responsible for time/frequency slot allocation. Hence the mobile does not see interference from other mobiles within the same cell. Assume that time is slotted and divided into frames. Each frame consists of N time slots, which are of equal length and can be used to transmit one fixed-length packet per slot. When packet i is generated at time t_i it announces a deadline d_i (in units of frames), so that if it is not successfully transmitted by time $e_i = t_i + d_i$ (the extinction time), the packet is considered lost and is dropped. We assume that all errors can be detected and the mobile receives an acknowledgement (ACK/NACK) immediately after the transmission of a packet. (The overhead for the acknowledgement transmission is ignored.) Any packet transmitted in error will be retransmitted until its deadline expires.

A. The Wireless Channel Model

We assume the variations in the wireless channel gain can be described by a finite-state discrete-time Markov chain (DTMC). Several authors have proposed Markov models for many different types of wireless channels [8,10,12]. The construction of these models follows a common procedure. The range of path gains are divided into a finite number of sections and each section becomes a state in the Markov chain. Then a transition matrix, which determines the probability of jumping from one state to another, is constructed in order to accurately model the channel characteristics (e.g. log-normal shadowing or Rayleigh fading). While we do not require any particular set of channel characteristics, we do require that the jumps of the DTMC occur at the time slot boundaries.

Let i represent one of the possible states in $\mathbf{I}$, where $\mathbf{I}$ is the set of states in the DTMC. The probability of jumping from state $i \in \mathbf{I}$ at time n to another state $i' \in \mathbf{I}$ at time $n+1$ is defined by the transition matrix $Q(i, i')$.

B. The Data Traffic Model

The traffic generated by the mobiles is also described by a DTMC. Each state in the traffic Markov Chain corresponds to a deadline and a number of packets. At the end of a frame a mobile will generate the number of packets and deadline corresponding to its state. A simple example of such a model is the popular ON/OFF two-state Markov Chain for voice traffic. In each time slot the chain can move from the OFF (resp. ON) state to the ON (resp. OFF) state with probability p (resp. q). If the chain is in the ON state at the end of a frame it will generate a packet with deadline d frames in the future. If it is in the OFF state it will generate nothing.

C. Implementation Specifics

Our proposed QoS algorithm is composed of two phases, a setup phase and an execution phase. During the setup phase we follow the method in the next section to solve the appropriate dynamic program. The solution of this dynamic program provides an optimal link adaptation policy for the power, coding, and modulation choices available to a mobile device. The resulting policy is optimal in that it minimizes the average transmission power subject to constraints on the probability of losing a packet and delay. The optimal control policy is stored in a look-up table, which is loaded onto the mobile. In the execution phase the mobile receives information from the base-station about the channel and combines this with its own information on traffic waiting for transmission. These pieces of information are then used to find the correct link adaptation policy in the look-up table.

III. Solution Method

The following section contains the dynamic programming formulation of our QoS algorithm. The first section develops a simple finite-horizon dynamic program. The ensuing sections develop extensions to this formulation in order to account for multiple traffic deadlines, Markov modulated traffic, and imperfect channel estimates.

A. Simple Dynamic Program Recursion

We begin with a simple finite-horizon dynamic program and two critical assumptions. First, assume packets arriving at the start of a frame have a transmission deadline at the end of the frame. Further assume the traffic Markov chain consists of only one state. The second assumption simplifies our initial notation. The first assumption ensures the link adaptation policy chosen in each frame is independent of the policy used in other frames. Since all data is flushed from the queue at the end of each frame (i.e. when the deadline expires), the link control in a frame will not affect the amount of data awaiting transmission in a following frame. Hence we can view each frame as an independent stochastic control problem. Although this assumption is common in the literature [1,3,6], it limits the data traffic to a very small class of models. We will relax both of these assumptions in the following sub-sections.

At the start of each frame the mobile generates a fixed number of bytes K that announce a common extinction time of $e_i = t_i + 1$, the end of the frame. In each time slot the mobile may choose an action $\vec{a} \in \mathbf{A}$, where $\mathbf{A}$ is the set of available transmitter power, coding, and modulation choices. The components of $\vec{a} = (a_p, a_b)$ are the selected transmitter power and the effective amount of data (in bytes) transmitted in one time slot when using coding and modulation choice a_b.

We shall consider a process observed at discrete time points $n = 0, 1, 2, 3, \ldots, N$, where $n = 0$ is the start of a frame and $n = N$ is the end. At each time point the state of the process $\vec{s} \in \mathbf{S}$ is a vector with components $\vec{s} = (s_d, i)$, the number of bytes waiting for transmission and the state of the wireless channel.

For any given channel state we assume the bit error probability (B_{err}) is known for a given modulation (GMSK, QPSK, etc.) and coding choice. In our case the quantity of interest is the packet error probability (P_{err}), which is a function of B_{err}, equalization, interleaving, error correcting codes, and other system-specific parameters. We assume P_{err} is known for a given action $\vec{a}$ and channel state i.

Let $\mathbf{F}$ be the set of all control policies. For any vector $\vec{f} \in \mathbf{F}$, let $f(\vec{s}, \vec{a}, n)$ denote the probability of choosing action $\vec{a}$ in state $\vec{s}$ at time n. For each action $\vec{a}$ the transmitter pays a penalty $r(\vec{a})$, the power used for transmission. Furthermore, if the mobile reaches time $n = N$ with $s > 0$ packets left in the queue, it pays a penalty $g(\vec{s})$. The purpose of the penalty $g(\vec{s})$ is to enforce some form of QoS in the mathematical formulation, the mobile does not actually pay the penalty in the execution phase. A common choice for this penalty is $g(\vec{s}) = \lambda s_d$, a linearly increasing function of the amount of data lost to deadline expiration. Several authors [5,6] have explored more complicated penalty functions (sometimes called utility functions) for various types of QoS constraints. We do not explore these functions here since, as we will show later, the penalty function can ultimately be replaced by an explicit performance constraint.

Define $V^n_{s_d,i}$ as the expected value of the total power consumed and penalty paid starting from time n in state $\vec{s} = (s_d, i)$. Our goal is to find a policy $\vec{f} \in F$ that minimizes $V^0_{K,i}$, call this policy $\vec{f}^*$. It is easy to show there exists an optimal Markov policy [2], $f^*(n, \vec{s}, \vec{a})$, that depends only on the current system state. This policy is created by choosing the optimal action $\vec{a}$, for each state $\vec{s}$ at every step of the following recursion:

$$\begin{aligned} V^n_{s_d,i} &= \min_{\vec{a} \in \mathbf{A}} \Big[r(\vec{a}) \\ &+ \sum_{i' \in \mathbf{I}} Q(i,i')(1 - P_{err}(\vec{a}, i')) V^{n+1}_{s_d - a_b, i'} \\ &+ Q(i,i') P_{err}(\vec{a}, i') V^{n+1}_{s_d, i'} \Big], \end{aligned} \tag{1}$$

with a terminal penalty vector

$$V^N_{s_d,i} = \lambda s_d. \tag{2}$$

B. Multiple Deadlines

We now relax the first assumption from the previous section and allow arriving data to have deadlines later than the current frame. We will model the data component of our state space as a vector $\vec{s_d}$. Each component of $\vec{s_d}$ represents the number of bytes awaiting transmission with different deadlines. For example, the first element of $\vec{s_d}$ is the number of bytes with a deadline of one frame in the future, the second element represents the number of bytes with a deadline of two frames in the future, and so forth. At the end of each frame all data with expired deadlines (i.e. the first element of $\vec{s_d}$) will be discarded. All other data will be moved up one element in $\vec{s_d}$ to denote that they are one frame closer to expiration. We keep the second assumption from the previous section and assume the mobile generates K bytes of data at the end of a frame, now with fixed deadline $d > 1$. Finally, we assume data with a deadline of k frames in the future will receive priority over data with a deadline of $k+1$ frames in the future (i.e. earliest deadline first queueing).

The addition of multiple deadlines to this formulation creates a significant problem. The link adaptation policy in one frame will now affect the state space of future frames. For example, suppose we choose a very low transmission power in the current frame and are unable to successfully transmit any bytes of data. If these bytes are then held over to the next frame, they will increase the delay of new arriving data. Therefore our finite horizon dynamic program, which assumes each frame is an independent problem, is no longer sufficient.

In order to correctly model the system we must convert our finite horizon model into an infinite horizon dynamic program. However, any infinite horizon control policy must be stationary [2]. Through a common trick in Markov chain analysis we can augment the problem state space with the time indices $n \in [0, 1, \ldots N]$. Therefore elements of the state space $\vec{s}$ will have components $(\vec{s_d}, i, n)$, the state of the data buffer, the state of the channel, and the current time slot. The component n will increment by one until it reaches N (i.e. the end of a frame), at which point n will jump back to zero (i.e. the start of a new frame). This state space augmentation will convert a non-stationary decision process into a stationary, periodic Markov decision process.

For $\vec{s} \in \mathbf{S}$ and $\vec{f} \in \mathbf{F}$ define the limiting average value of the stationary policy $\vec{f}$ as

$$V(\vec{f}) = \limsup_{m \to \infty} \left[\left(\frac{1}{m+1} \right) \sum_{k=0}^{m} E_{\vec{s}\vec{f}} \left[r(\vec{s}, \vec{f}, k) \right] \right], \tag{3}$$

where $r(\vec{s}, \vec{f}, k)$ is the reward at time k resulting from the policy $\vec{f}$ in state $\vec{s}$.

Our goal is to find an $\vec{f} \in \mathbf{F}$ that minimizes $V(\vec{f})$. From [2] we know this problem can be solved through the following linear program.

$$\min_{x} \sum_{\vec{s} \in \mathbf{S}} \sum_{\vec{a} \in \mathbf{A}} r(\vec{s}, \vec{a}) x_{\vec{s}\vec{a}} \tag{4}$$

subject to:

$$\sum_{\vec{s} \in \mathbf{S}} \sum_{\vec{a} \in \mathbf{A}} \left(\delta(\vec{s}, \vec{s}') - p(\vec{s}' | \vec{s}, \vec{a}) \right) x_{\vec{s}\vec{a}} = 0, \ \vec{s}' \in \mathbf{S} \tag{5}$$

$$\sum_{\vec{s} \in \mathbf{S}} \sum_{\vec{a} \in \mathbf{A}} x_{\vec{s}\vec{a}} = 1, \tag{6}$$

$$x_{\vec{s}\vec{a}} \geq 0; \ \vec{a} \in \mathbf{A}, \ \vec{\mathbf{s}} \in \mathbf{S},$$

where $\delta(\vec{s}, \vec{s}')$ is the Kronecker delta, $x_{\vec{s}\vec{a}}$ is the probability of taking action $\vec{a}$ in state $\vec{s}$, and $p(\vec{s}' | s, \vec{a})$ is the probability of jumping to state $\vec{s}'$ given action $\vec{a}$ in state $\vec{s}$. The state-action frequencies $x_{\vec{s}\vec{a}}$ provide a unique mapping to an optimal strategy $\vec{f}^*$.

Recall that packets with expired deadlines are dropped from the data buffer at the end of each frame. Therefore there exists a subset of the state space, $\tilde{\mathbf{S}} \subset \mathbf{S}$, corresponding to the set of states where a packet is dropped due to deadline expiration. In order to constrain the probability of packet loss we add the following constraint to (4),

$$\sum_{\vec{a} \in \mathbf{A}} \sum_{\vec{s} \in \tilde{\mathbf{S}}} x_{\vec{s}\vec{a}} \leq \alpha. \tag{7}$$

We can further define sets of states corresponding to the loss of any number of consecutive packets. In some cases an expansion of the state space may be required in order to track the recent packet loss history. Each additional constraint on the distribution of packet loss requires an extra linear constraint in (4). Also notice that we no longer require the penalty function for dropping a packet since the probability of packet loss is explicitly constrained.

C. Markov Modulated Traffic

Until now we have assumed a fixed amount of data, K, arrives at the transmitter at the end of each frame with a fixed deadline of $d > 1$ frames in the future. Though convenient, this assumption is only appropriate for a very small class of traffic models. As we mentioned earlier, the statistics of a much more general class of traffic can be described by a finite state Markov chain. Each state in the traffic Markov chain corresponds to a number of bytes and a deadline. At the end of each frame, the number of bytes corresponding to the traffic state are added to the buffer with a transmission deadline also determined by the state. There will be a matrix describing the transition probabilities for the traffic chain, and this matrix is incorporated into the dynamic program recursion (or into the linear program) in exactly the same fashion as the matrix that describes the transitions for the wireless channel.

D. Imperfect Channel Estimates

We now relax our assumption of perfect channel estimates. In mobile wireless systems, channel estimates are computed at the base station and forwarded to the mobile on a regular basis. This process will introduce both error and delay in the estimates. We first consider the estimation error. Due to the random nature of the channel, it is impossible for the base station to precisely determine the state of the channel. The best estimate a base station can provide is a probability distribution over the possible channel states. Consider our original Markov model for the wireless channel where we assume the channel parameters can only take values from a finite set $\mathbf{I}$. More accurately, the channel estimation algorithm discretizes the channel into a finite set. Furthermore, suppose the base station transmits an index j from the set of possible estimates $\mathbf{J}$, where each $j \in \mathbf{J}$ now corresponds to a discrete probability distribution over the set $\mathbf{I}$. This change does not add any computational complexity to the dynamic program; we are simply exchanging one set of indices for another. However, we must account for this change when computing the probability of packet error. We defined the packet error probability as $P_{err}(\vec{a}, i)$, a function of the action choice $\vec{a}$, and channel state i. Since we now only have a distribution of the channel state, $\mathbf{i}$ is a random variable; hence $\mathbf{P_{err}}(\vec{\mathbf{a}}, \mathbf{i})$ is also random. The probability of packet error as a function of a channel estimate $j \in \mathbf{J}$ is the expectation of the random variable $\mathbf{P_{err}}(\vec{\mathbf{a}}, \mathbf{i})$ with respect to the distribution defined by j. If we define $P_{err}(\vec{a}, j)$ as the probability of packet error for an action choice $\vec{a}$ and a channel estimate j, we can compute its value as

$$P_{err}(\vec{a}, j) = \sum_{k \in \mathbf{I}} P(i = k | j) \ P_{err}(\vec{a}, k), \tag{8}$$

where $P(i = k | j)$ is the probability the channel is in a particular state $k \in \mathbf{I}$ given a received estimate j.

We also account for the delay in channel estimates. Assume the base station only transmits channel estimates once every m frames. The accuracy of the estimate will degrade over time. If we know the statistics describing the channel evolution (e.g. the transition matrix of the channel Markov chain), we can determine the distribution of the channel state at any time using only an initial distribution and the number of time slots since that distribution was received. For example, suppose we receive the estimate j from the base station at time $n = 0$ that corresponds to a discrete distribution $\vec{\lambda_0}$ (a row vector of probabilities). At time $n \leq m$ our best estimate of the distribution of the channel state is

$$\vec{\lambda_n} = \vec{\lambda_0} Q^n, \tag{9}$$

where Q is the transition matrix for the channel. Therefore at any time we can compute the probability of packet error using the formula in (8), with a base station estimate j and the time since that estimate was received.

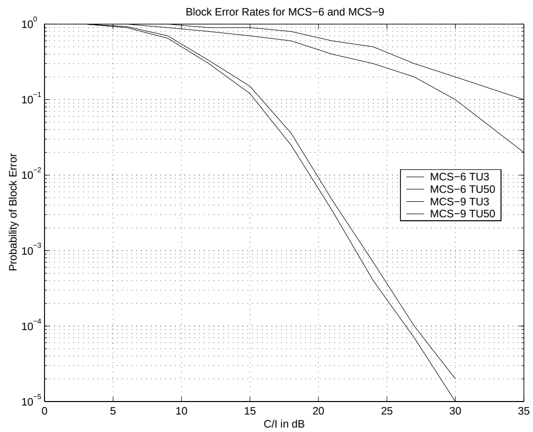

Fig. 1. Packet Error Probabilities for MCS-6 and MCS-9 in TU3 and TU50

IV. Numerical Evaluation

In this section we will present the results of a simple system level simulation of the GSM EDGE Radio Access Network (GERAN).

A. Simulation Setup

A standard EDGE frame consists of eight time slots lasting a total of 20 milliseconds. We consider a one slot EDGE mobile transmitting deadline sensitive data. We would like to guarantee 99% of the packets will meet their transmission deadlines. For each packet the mobile may select a transmitter power between 0.02 and 0.8 Watts, in increments of 2dB. The mobile may also choose either the MCS-6 or MCS-9 coding scheme. These modulation and coding schemes are two of the five enhanced 8PSK data rates for EDGE. MCS-6 is rate $\frac{1}{2}$ convolutionally encoded 8PSK and can transmit 74 bytes per 20ms block (i.e. 29.6 Kbps). MCS-9 is full rate 8PSK and can transmit 148 bytes per 20ms block (i.e. 59.2 Kbps).

We assume the data traffic arriving at the mobile follows the ON/OFF DTMC described in Section 2. If the chain is in the ON state at the end of a frame the mobile will generate one 74 byte packet with a deadline of 3 frames (i.e. 60 ms) in the future.

B. The Wireless Channel

We used data available at the 3GPP website (www.3GPP.org) for the packet error performance of EDGE [5] in a typical urban multipath environment, a channel model in the ETSI COST 207 specification [3]. The 3km/h and 50km/h versions of this channel model (TU3 and TU50 respectively) are often used to evaluate the performance of GSM and EDGE. The packet error probabilities for the MCS-6 and MCS-9 coding schemes as a function of received Signal to Interference Ratio are plotted in figure 1.

We model the free space path loss and shadowing as a DTMC. As suggested by Gudmundson [8], we model the log-normal shadowing as a Gaussian process in dB units. In order to fully describe the shadowing model we must specify a mean path loss, standard deviation, and a correlation coefficient. Once these parameters are determined we split the range of channel gains into a finite number of bins and construct a Discrete Time Markov Chain that approximates the Gaussian process using the method suggested in [12]. The following table contains the channel gain regions for the DTMC and other channel parameters.

Path Loss Ranges in dB	$(-\infty$,112.5),(112.5,117.5) (117.5,122.5),(122.5,127.5) (132.5,∞)
Mean Path Loss	120dB
Shadowing Std. Dev	$\sigma = 7dB$ macrocell model $\sigma = 4.3dB$ microcell model
Macrocell Shadowing Correlation	$\zeta_D = .82$ for $D = 100m$
Microcell Shadowing Correlation	$\zeta_D = .3$ for $D = 20m$
Noise Threshold N_0	-150dBW
Carrier Frequency	900MHz

We consider two shadowing models [8], one for typical macrocells, and one for typical microcells. In the macrocell model, the log-normal shadowing is highly correlated over fairly large distances (100 meters), whereas the microcell shadowing de-correlates over relatively short distances (20 meters). Therefore, if a channel estimate is delayed, we expect the accuracy of that estimate to degrade more quickly in the microcell channel, especially if the mobile is moving at the higher speed of 50km/h.

C. Channel Estimates

In a standard EDGE mobile network, the base station passes power control commands to the mobile once every 480ms. In our formulation the base station passes the channel measurements rather than the power control commands, thereby allowing the mobile to perform its own link adaptation.

A recent study [5] considered the impact of shortening the power control interval on system performance. In particular, intervals of 20ms, 40ms, 60ms, 120ms, 240ms, and 480ms were considered. The study concluded there was no improvement in system performance by decreasing the update interval below 480ms for a 3km/h mobile or 120ms for a 50km/h mobile. However, this study only considered a macrocell shadowing environment. In a microcell, where the shadowing changes much more quickly, these update intervals may not be sufficient. We considered all 6 of the above update intervals as well as the two shadowing models in order to answer this question.

D. An Algorithm for Comparison

As a comparison for our dynamic program we will consider the standard Fixed-SIR power control algorithm. For this example our goal is to keep the probability of packet loss below 0.01. In order to meet this QoS constraint, the appropriate SIRs for each mobile speed, shadowing model, and estimate delay were determined by simulation. The graph in Figure 4 shows the average power consumed per transmitted packet versus estimate delay for the Fixed-SIR policies. Clearly the delay in channel estimates will have a significant impact on power consumption when the channel is changing rapidly (e.g. high speed mobiles and microcells). In fact, with the standard estimate delay of 480ms, a high speed mobile in a microcell will consume almost twice the power of a slow moving mobile in a macrocell in order to satisfy the same QoS constraint.

E. Performance Evaluation

We now consider the performance of the proposed Dynamic Programming algorithm. Figures 2 and 3 plot several power control policies generated by our algorithm. The curves in Figure 2 show the optimal power choice as a function of channel and buffer state for a 50km/h mobile in a microcell shadowing environment. Figure 3 shows the same plots for a 3km/h mobile. The shaded region in each plot represents the states where the MCS-9 code scheme (i.e. uncoded 8PSK) is used. In all other states MCS-6 (i.e. rate $\frac{1}{2}$ coded 8psk) is used. The six graphs in each figure represent one of the possible estimate delays (i.e. 20ms, 40ms, 60ms, 120ms, 240ms, and 480ms). The channel states are ordered from best to worst quality, and the buffer states are ordered first by the number of packets queued and then by packet urgency. For example, buffer state 1 corresponds to one queued packet with 3 frames left before expiration, buffer state 2 represents 1 packet queued with 2 frames before expiration, etc.

First, consider the plot for 20ms of estimate delay (the top-left plot) in each figure. In this case the mobile receives estimates quite frequently and has accurate information about the state of the channel. When the channel quality is poor the mobile will decrease transmit power if its buffer does not contain a packet that is about to expire. If the mobile does have a packet that will expire at the end of the current frame it raises its transmit power to overcome the poor channel. For example, buffer state 4 (2 packets with deadlines of 2 and 3 frames in the future) requires a lower transmit power than state 3 (only one packet, but with a deadline of 1 frame in the future). Notice that the MCS-9 code is only used in the best channel states and when there is a large backlog of data. Recall that MCS-9 can transmit twice as much data as MCS-6, but it requires an extremely high signal-to-interference ratio for succesful transmission.

Now consider the remaining plots with estimate delay increasing to 480ms at the bottom-right corner of each figure. For a 50km/h mobile in a microcell, the probability the channel remains in the same state over large time intervals is quite low. Therefore the mobile will attempt to take advantage of a good channel by using the MCS-9 code. Notice that a large update delay renders the channel estimate virtually useless. Furthermore, as estimate delay increases the mobile must raise its transmit power substantially for the MCS-9 code. Indeed, MCS-9 becomes too expensive in terms of power when the estimate delay reaches 480ms.

In the plots for the 3km/h mobile we see that the control policy is relatively insensitive to delay. Once again this is due to the nature of the shadowing process. Even the large update interval of 480ms is sufficient to accurately track the channel state. Notice the MCS-9 code is used less often than in the 50km/h case. Since the channel is likely to remain in a high quality state for a long period of time, there is less incentive for the mobile to take advantage of a good channel state and waste large amounts of power on a MCS-9 packet.

If we examine the plot of average power consumption in Figure 5 our Dynamic Programming algorithm appears to overcome the shortfalls of the Fixed-BER algorithm. In fact, our Dynamic Programming algorithm does not appear to incur a penalty as estimate delay increases. Obviously this can not be the case since it implies the channel updates are not useful. As it turns out, our Dynamic Programming formulation with only a single probability constraint will perform progressively worse, in terms of consecutive packet loss, as estimate delay increases. In order to correct this problem we add a second constraint, which forces the probability of losing 2 consecutive packets to be less than .0001.

Figure 6 plots the average power per packet for a 50km/h mobile in a microcell shadowing environment using the Fixed-SIR policy and both Dynamic Programming formulations. Notice the power consumption for the Dynamic Program with the extra constraint increases as estimate delay increases. Average power plots for the other channels (3km/h microcell/macrocell and 50km/h macrocell) with the new constraint also have a similar shape, though the penalty for estimate delay in these cases is not as high. After adding the extra constraint on consecutive packet loss, our algorithm still consumes less power than the Fixed-SIR policy. Though important, this gain in power consumption is not necessarily the most significant feature of Figure 6. The power consumption of the Fixed-SIR policy (Figure 4) for a 50km/h mobile in a Microcell increases sharply for estimate delays greater than 40ms. In our proposed algorithm, the average transmit power does not increase markedly until the estimate delay exceeds 120ms. From the perspective of system design and implementation, this "gain" in estimate delay could greatly simplify command and control protocols as well as the channel update algorithm.

V. Conclusion

The need for deadline based QoS in wireless channels is clear: the increasing demand for multimedia traffic over packet data networks and the future growth of wireless network access require this form of QoS. We have demonstrated that it is possible to provide robust QoS schemes for delay sensitive data in erratic transmission environments using appropriate power, coding, and scheduling policies. Our control algorithm allows for complex QoS constraints, such as a constraint on probabilities of consecutive packet loss. Furthermore, we provide the means to explicitly account for channel estimation error and estimate delay. In contrast, the conventional Fixed-SIR scheme does not take advantage of fluctuations in the channel or traffic to save energy, nor does it account for inaccurate channel estimates. The method of power control introduced here adapts to the erratic behavior of the communication channel and accounts for the uncertainty in the system in order to meet QoS goals and minimize power consumption. The next step in the development of this QoS algorithm is a much larger scale solution for GSM EDGE. In particular, we plan to implement the full set of coding schemes, Incremental Redundancy, Adaptive Multi-Rate voice coders, and allow for more complex models of user mobility. This formulation can also be extended to interference based systems (e.g. CDMA) using multiple controller dynamic programming [13]. The increased complexity resulting from multiple controllers requires us to reserve this topic for a journal paper.

Though our results indicate the promising possibility of QoS guarantees for delay constrained communication in an unreliable environment, there are additional issues. For example, it is clear that the channel model needs additional refinement. As we demonstrated earlier, the optimal control policy is heavily dependent on both the speed of the mobile and the nature of the shadowing environment. In cellular systems we can expect mobiles to attain speeds higher than 50km/h and there are many other shadowing models worthy of investigation. Further research examining the impact of high speed mobiles and alternate channel models is necessary.

References

[1] N. Bambos, S. Kandukuri, *Power Controlled Multiple Access in Wireless Communications Networks*, IEEE INFOCOM 2000.

[2] D. Bertsekas, *Dynamic Programming*, Prentice Hall, June 1987.

[3] COST 207 Management Committee, *COST 207: Digital Land Mobile Radio Communications*, Commission of the European Communities, 1989.

[4] M. Elaoud, P. Ramanatan, *Adaptive Allocation of CDMA Resources for Network-level QoS Assurances*, Mobicom 2000.

[5] Ericsson Wireless, *System Performance Impact of Power Control Interval*. 3GPP TSG GERAN Tdoc 2g00-061, Helsinki, Finland, August 2000.

[6] D. Famolari, D. Goodman, N. Mandayam, V. Shah *A New Framework for Power Control in Wireless Data Networks: Games, Utility and Pricing*, Wireless Multimedia Technologies, Kluwer Academic Publishers, 1999.

[7] N. Feng, N.B. Mandayam, D.J. Goodman, *Joint Power and Rate Optimization for Wireless Data Services Based on Utility Functions*, CISS'99.

[8] M. Gunmundson, Correlation Model for Shadow Fading in Mobile Radio Systems, Electronics Letters, Vol. 27, November 1991.

[9] F. Gunnarsson, Fredrik Gustafsson, *Power Control with Time Delay Compensation*, VTC'00, Fall 2000.

[10] Y.Y. Kim, S. Li, *Modeling Fast Fading Channel Dynamics for Packet Data Performance Analysis*, Proceedings IEEE INFOCOM '98.

[11] J. Rulnick, N. Bambos, *Mobile Power Management for Wireless Communications Networks*, Wireless Networks, Vol. 3, 1997.

[12] G. Stuber, Principles of Mobile Communication, Kluwer Academic Publishers, 1999.

[13] K. Vrieze, J. Filar, Competitive Markov Decision Processes, Springer Press, 1997.

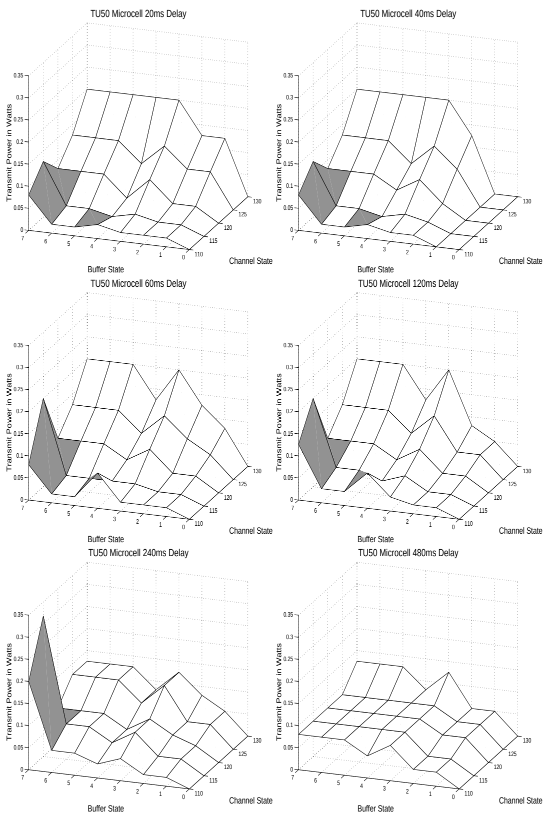

Fig. 2. Power vs. Buffer State and Estimate Delay for the TU50 Microcell. The MCS-9 code is used in the shaded region, otherwise MCS-6 is used.

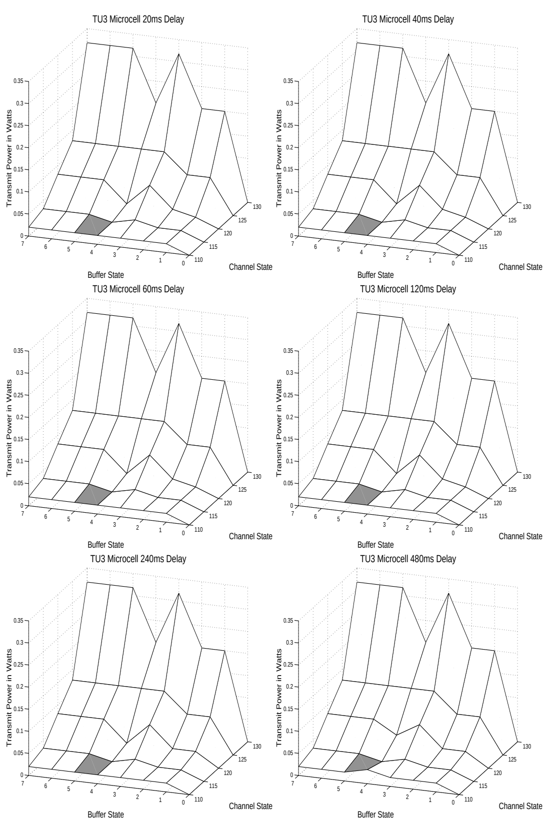

Fig. 3. Power vs. Buffer State and Estimate Delay for the TU3 Microcell The MCS-9 code is used in the shaded region, otherwise MCS-6 is used.

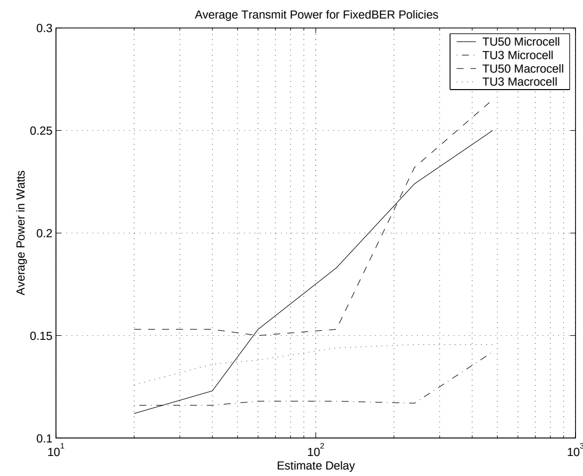

Fig. 4. Average Power per Packet vs. Estimate Delay for the Fixed-SIR Control Scheme

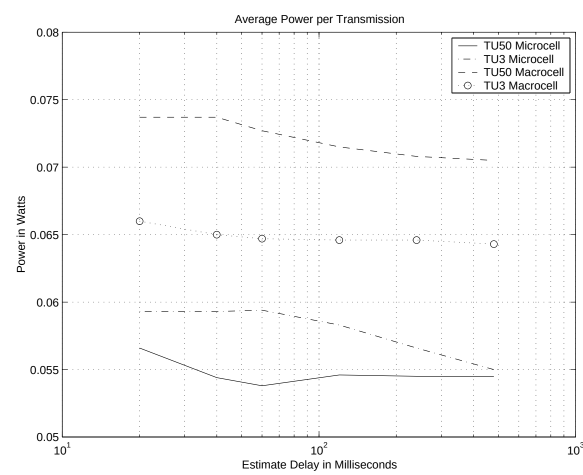

Fig. 5. Average Transmit Power vs. Estimate Delay for the Dynamic Programming Algorithm

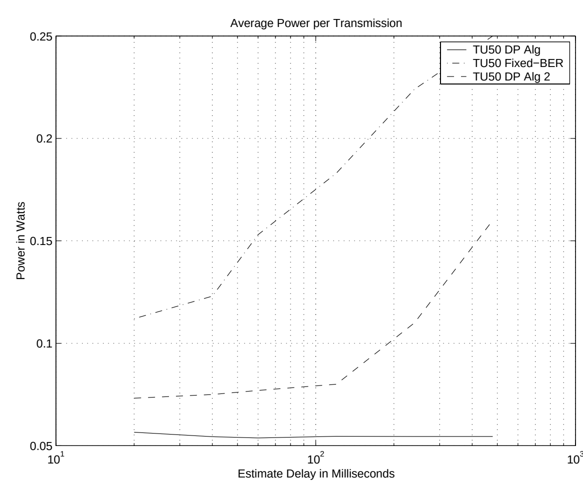

Fig. 6. Average Transmit Power vs. Estimate Delay for the TU50 Microcell: DP Algorithm, Modified DP algorithm, and Fixed-BER policy