QoS Control for WCDMA High Speed Packet Data

Patrick A. Hosein
Ericsson Wireless Communications Inc., 6455 Lusk Boulevard, San Diego, CA 92121, USA
hosein@ericsson.com

Abstract – Wideband CDMA Release 5 is expected to support peak downlink bit rates of 10 Mbps for use with bandwidth intensive data and multimedia applications. Such rates are achieved through fast link adaptation, fast Hybrid ARQ and fast scheduling over a shared forward link packet data channel. Total throughput can be maximized by having the frame scheduler take into account the instantaneous radio conditions of users and serving users during their good radio condition periods. This results in high user diversity gains. However in order to support QoS guarantees, it will sometimes be necessary to serve users experiencing bad radio conditions in order to maintain their requested QoS levels. In this paper we present a flexible algorithm that provides user QoS guarantees while at the same time achieving some user diversity gains.

Keywords: WCDMA; HSDPA; QoS; Scheduler; Wireless; 3G.

1. INTRODUCTION

Wideband CDMA (WCDMA) Release 5 is expected to support peak downlink bit rates of 10 Mbps. This will be achieved by introducing fast link adaptation, fast Hybrid ARQ and fast scheduling over a shared forward link packet data channel (see [1], [2] and [6] for details). However, in addition to increased performance, there is a growing need for QoS support for user applications. In this paper we address the issue of scheduling over this shared packet data channel with the objective of providing per user QoS demands.

Although various schedulers have been proposed for scheduling users over a shared packet data channel, many either do not provide QoS guarantees or are relatively complicated to implement. Our proposed algorithm maintains the simplicity of the Proportional Fair Scheduler while allowing the operator to trade-off how closely QoS guarantees are maintained with the user diversity gains achievable by taking advantage of channel conditions. The scheduler attempts (it may not always be possible to provide a user’s QoS demands because of radio conditions) to satisfy the QoS demands of each connection and maximizes throughput (or any other objective) over any excess capacity.

In the next section we provide a mathematical formulation of the scheduling algorithm and show that the optimal solution can be obtained with a gradient ascent method. We then introduce the notion of Barrier Functions which we use to enforce the QoS demands. Finally we illustrate the effectiveness of our approach with examples.

2. MATHEMATICAL FORMULATION OF THE SCHEDULING PROBLEM

Kelly [5] provides a formulation of this shared resource allocation problem as follows:

maximize \[ F(\vec{r}) \equiv \sum_{i=1}^{k} U_i(r_i) \]  
subject to \[ \sum_{i=1}^{k} r_i < C \]  
over \[ r_i \geq 0, \ 1 \leq i \leq k. \]

where, \( k \) = the number of active users competing for the shared channel, \( r_i \) = the average throughput of user \( i \), \( C \) = the channel capacity, \( U_i(r_i) \) = the utility function of user \( i \).

If we assume the utility function of each user is strictly concave and differentiable, then the same also holds for the objective function \( F \). Since the feasible region is compact then an optimal solution exists, is unique and can be found by Lagrangian Methods. The optimal set of rates, which we denote by \( \{r_i^*\} \), has the property that \( U_i^*(r_i^*) \) is the same for all users. Note that this optimal point can be computed explicitly.

The optimal solution changes over time since the channel capacity and the number of active users both vary with time (hence we are aiming at a moving target). Furthermore, although the optimal throughputs can be computed, the system cannot be instantly moved to that allocation. One can only move towards the optimum by making appropriate scheduling decisions. Therefore the best that one can do is to continuously move towards the “present” optimal solution by moving in the direction of steepest ascent at each decision point. Therefore the user, who if served, results in movement along the maximum objective function gradient direction is chosen. Assume that (exponentially smoothed) user throughputs are computed as follows:

\[ r_i(n+1) = \begin{cases} (1 - \frac{1}{\tau})r_i(n) + \frac{d_i(n)}{\tau} & \text{if } i \text{ served in slot } n, \\ (1 - \frac{1}{\tau})r_i(n) & \text{otherwise} \end{cases} \]  

(4)
where \( d_i(n) \) is the bit rate of user \( i \) if served during the \( n \)th period, \( \tau \) is the time constant of the smoothing filter and \( r_i(n) \) denotes the average throughput that was previously computed. Denote the gradient of the utility function of user \( i \) at the start of the \( n \)th frame by \( U'_i(r_i(n)) \) and let \( F'_j(\vec{r}(n)) \) denote the gradient of the objective function in the direction of serving user \( j \). The standard dual-ascent algorithm would determine the value of \( j \) with the largest gradient and move to the maximal point along that direction. However in this case such arbitrary changes are not possible. If a user is chosen then that user is served and the throughputs are updated. This determines the amount of movement in the direction corresponding to serving the \( j \)th user. Therefore, the dual-ascent algorithm reduces to finding the maximum gradient direction and serving the corresponding user.

Let us parameterize the movement along the ray corresponding to serving user \( j \) by \( \alpha \). The objective \( F_j \) as a function of \( \alpha \) can be written as,

\[
F_j(\alpha) = \sum_{i=1}^{k} U_i(r_i(n) + \alpha(r_i(n + 1) - r_i(n))).
\]

Taking the derivative with respect to \( \alpha \) and evaluating the derivative at \( \alpha = 0 \) we get:

\[
F'_j = U'_i(r_j(n)) \frac{d_j(n) - r_j(n)}{\tau} - \sum_{i=1,i\neq j}^{k} U'_i(r_i(n)) \frac{r_i(n)}{\tau}.
\]

We can re-write this to obtain the gradient in the direction corresponding to serving user \( j \) as:

\[
F'_j(\vec{r}(n)) = U'_j(r_j(n)) \frac{d_j(n) - r_j(n)}{\tau} - \sum_{i=1}^{k} U'_i(r_i(n)) \frac{r_i(n)}{\tau}.
\]

Therefore the maximum gradient direction is given by,

\[
j^* = \arg \max_j \{ F'_j(\vec{r}(n)) \} = \arg \max_j \{ d_j(n)U'_j(r_j(n)) \}
\]

since the summation term is common to all directions and so can be removed.

Assume that the provider charges based on the number of bits delivered to the user. In this case the utility function is a linear function of the average throughout of a user and hence \( U(r) = \alpha r \) for some constant \( \alpha \). The resulting scheduler will pick the user for which:

\[
j^* = \arg \max_j \{ d_j(n) \}
\]

which essentially means pick the user with the highest achievable bit rate. This results in maximum sector throughput and hence maximum revenue. However, this scheduler, (the max CIR scheduler) will always serve those users who are in good conditions and will rarely serve those who are in bad conditions which leads to unfair allocation of resources.

In order to address the unfairness of the CIR scheduler, the Proportional Fair utility function is sometimes used. In this case the utility function is given by \( U(r) = \log(r) \) and hence the resulting scheduler picks the user such that:

\[
j^* = \arg \max_j \left\{ \frac{d_j(n)}{r_j(n)} \right\},
\]

This provides some degree of fairness at the expense of reduced sector throughput (since users in bad conditions must also be served).

In order to provide QoS guarantees as well as take advantage of user diversity gains, we introduce the notion of a Barrier Function in the next section. The resulting scheduling problem is again solved using a dual-ascent approach. Note that one could also use Primal Space methods as done in [3]. The Primal Space approach provides additional flexibility in the choice of Barrier Functions (in particular, they need not be differentiable).

3. Barrier Functions for QoS Constraints

The QoS demands of an application will place throughput, delay, jitter and packet loss rate bounds on the underlying connection. In this paper we focus on the throughput constraints but the approach can be applied to the other constraints as well. Recall that the utility function of a user is a function of the average throughput. If we ignore the QoS constraints then a utility function of \( U(r) = r \) provides the maximum sector throughput. We use this as our primary utility function (others, such as the Proportional Fair utility function, may also be used).

Assume that a minimum throughput must be maintained for a particular connection. Instead of imposing a hard constraint (i.e., immediately reacting so that connections that violate this constraint are immediately served to move them back into the feasible region) we introduce Barrier Functions to reduce movement outside of the feasible region. For example, consider the case where a minimum throughput of \( r_{\text{min}} \) is required. Consider the following utility function:

\[
U(r) = r + (1 - e^{-\beta(r-r_{\text{min}})}).
\]

As long as the throughput exceeds the minimum value then this is approximately the max CIR function. When \( r \) drops below the minimum value then the utility function rapidly decreases. Hence the scheduler would be forced to serve users with throughputs below the minimum (since the gradient rapidly increases). The parameter \( \beta \) determines the rate at which the the penalty for violating the constraint increases. This value should be based on the Service Class of the user (e.g., Gold, Silver and Bronze.
services). Such functions are called Barrier Functions since they serve to perform a flexible “barrier” around the feasible region.

Figure 1 provides an example of this utility function for different values of $\beta$. Each user requires a minimum throughput of 60 kbps. As long as bandwidth is available then this requirement can be satisfied for all users. If not, a small penalty is incurred for violating the throughput constraints of Bronze users but a much larger penalty is incurred for Gold users. Hence Gold users continue to be given higher scheduling priority over the Silver users even when insufficient resources are available. In this case the gradient ascent algorithm picks the user for which $d U'(r)$ is maximal where

$$U'(r) = 1 + \beta e^{-\beta(r-r_{\text{min}})}.$$ 

We can similarly determine appropriate barrier functions for delay constraints, jitter constraints and queue length (buffer occupancy) constraints. The sum of these barrier functions plus the primary utility function will constitute the user’s overall utility function.

4. SIMULATION MODEL FOR THE WCDMA HSDPA

WCDMA Release 5 introduces a new transport channel, the HS-DSCH (High Speed Downlink Shared Channel) for multiplexing user data. This channel incorporates three principles, fast link adaptation (adaptation of the transmission modulation and code to instantaneous channel conditions), fast Hybrid ARQ (frame re-transmissions and soft combining) and fast scheduling (scheduling decision is made after each frame transmission and may be based on users’ radio conditions).

Since our focus is on the frame scheduler, we consider a simplified model that takes into account the relevant aspects. We assume that each UE (User Equipment) reports its radio conditions (i.e., Carrier to Interference Ratio (CIR) values) to the Base Transceiver Station (BTS) at 3.33s intervals. The CIR value of each UE is mapped into one of 7 Modulation Coding Schemes (MCS). At 3.33s intervals, the BTS scheduler determines which UE is to be served next and transmits a single frame to the chosen UE using the corresponding MCS for that UE. We refer to the 3.33ms interval as the frame size and also the scheduling interval.

Since the user diversity gains are due to fast fading of the UE then only fast fading is modeled. We do this with a Discrete Time Finite State Markov Chain as follows. Each state corresponds to one of the possible MCS levels. We also assume a direct mapping between the CIR values and the chosen MCS (i.e., we do not consider the other factors that may affect the mapping). The Markov Model is run for each user to determine the UE’s channel conditions (and hence appropriate MCS) at each decision point. The resulting throughputs are discounted to take into account frame errors and re-transmissions.

Let $S = s_1, s_2, \ldots, s_K$ denote the state space of the corresponding $K$-State Markov chain and let $\Gamma_k$ represent the lower CIR threshold corresponding to state $s_k$. Denote the state transition period by $\tau$. For multi-path propagation environments, the received signal envelope has a Rayleigh distribution. Assuming additive Gaussian noise, the instantaneous CIR, $\gamma$, is distributed exponentially with probability density function

$$p(\gamma) = \frac{1}{\gamma_0}e^{-\gamma/\gamma_0}, \quad \gamma \geq 0,$$

where $\gamma_0$ is the average CIR. Given the distribution of the received CIR we can compute the steady-state distributions as follows:

$$\pi_k = \int_{\Gamma_k}^{\Gamma_{k+1}} p(\gamma) d\gamma = e^{-\frac{\Gamma_k}{\gamma_0}} - e^{-\frac{\Gamma_{k+1}}{\gamma_0}}, \quad k = 1, \ldots, K.$$ 

Let $f_m$ denote the maximum Doppler frequency defined as $f_m = v/\lambda$, where $v$ is the speed of the UE and $\lambda$ is the wavelength of the carrier signal. It can be shown (see [4]) that the rate at which the received CIR crosses a given threshold, $\Gamma$, in one direction is given by

$$N(\Gamma) = \sqrt{2\pi \Gamma \gamma_0 f_m e^{-\frac{\Gamma}{\gamma_0}}}.$$ 

One can then show (see [4] for details) that the transition probability can be approximated as follows:

$$P_{k,k+1} \approx \frac{N(\Gamma_{k+1})}{\pi_k}, \quad \text{and} \quad P_{k+1,k} \approx \frac{N(\Gamma_k)}{\pi_k}.$$ 


1. Hence we assume a 3 slot HSDPA TTI.
Finally, the probability of returning to the same state is given by:

\[ P_{k,k} = 1 - P_{k,k+1} - P_{k+1,k}, \quad k = 2, \ldots, K - 1, \]

while \( P_{1,1} = 1 - P_{1,2} \) and \( P_{K,K} = 1 - P_{K,K-1} \). This assumes that the transition period is short enough so that the probability of moving from one state to a non-adjacent state is negligible.

Let \( P(\tau) \) denote the transition probability matrix for a transition period \( \tau \). The transition probability matrix for transitions made over a period of time \( T \) can be obtained by taking \( P(\tau) \) to the power of \( T/\tau \). Let \( P(T) \) denote this matrix in the limit as \( \tau \) goes to zero,

\[ P(T) = \lim_{\tau \to 0} P(\tau)^{T/\tau}. \]

We obtain the state transition probabilities by solving for this matrix with \( T = 3.33 \text{ms} \). Note that this model has been shown to compare well with a Jakes simulation. AWGN link performance data were taken from [6].

5. Simulation Results

In order to illustrate the effectiveness of the proposed scheduler, we simulated the following problem. We assume a total of 24 (3 Kmph pedestrian) UEs and that each UE always has data for transmission on the forward link. We assume 12 Silver users and 12 Gold users and that the Silver users are in better radio conditions than the Gold users (i.e., the average CIR value of the Silver users exceeds that of the Gold users by 4 dB). Suppose that we wish to provide the Gold and Silver users with Circuit Switched Equivalent (CSE) rates of 128 kbps and 64 kbps respectively. Note that under these circumstances both the max CIR scheduler and the Proportional Fair scheduler would provide the Silver users with a higher throughput than the Gold users since they are in better radio conditions. Hence they cannot be used to provide the QoS guarantees.

In this particular case we will use a quadratic Barrier Function. This function will impose penalties for moving away from the desired CSE rates. We use the max CIR function as the primary utility function and hence the overall utility function is given by,

\[ U(r) = r - \beta(r - r_{\text{cse}})^2. \]

We use \( \beta = 0.02 \) for the Gold users and \( \beta = 0.05 \) for the Silver users (these determine the degree of compliance to the QoS demands). In the case of Gold users we have \( r_{\text{cse}} = 128 \) and for the Silver users we have \( r_{\text{cse}} = 64 \). The resulting scheduler thus chooses the UE for which:

\[ j^* = \arg \max_j \{ d_j(n)[1 - 2\beta(r_j(n) - r_{\text{cse}})] \}. \]

As long as resources are available then all users will achieve their CSE throughputs. If this is not the case then all users will maintain throughputs less than their desired CSE. The average throughputs within a class will be the same. The relationship of the throughputs between the classes can be obtained by equating their gradients.

In Figure 2 we plot the throughput for one Gold and one Silver user. We also plot the exponentially smoothed throughput that would be achieved by a constant bit rate user with the corresponding CSE rates. Note how closely the Gold and Silver throughputs compare with constant bit rate users. Furthermore, note that this holds true for all 24 users (but space does not allow for all plots). We repeated the same scenario but this time used the Proportional Fair scheduler. In Figure 3 we plot the throughputs for the same two users. We also plot the 128 kbps CSE case for comparison. Firstly, note that the Silver users experience higher throughputs than the Gold users. Furthermore, the Gold users do not achieve their desired CSE rates (although the Silver users far exceed theirs).

Note the following additional characteristic of the Barrier Function throughput results. The throughput of a user is limited from above and below the desired CSE by the penalty function. This results in a lower throughput variance than that of either max CIR or Proportional Fair. This has a positive effect on the TCP layer since there is less variance in the estimate of the round trip delay and also less likelihood of buffer overflows (and hence packet loss).

Another interesting property is the initial throughput experienced by new users. Suppose that 23 (11 Gold and 12 Silver) users have been active for a while and a new Gold user enters the system. Since its initial throughput is low it will be given high priority and served almost exclusively until its throughput reaches the CSE value. In Figure 4 we plot the through-
puts for the new Gold user together with the throughput for the corresponding CSE user. Note the high initial throughput experienced. This property implies that latencies for short packet transmissions (e.g., WAP, IM etc.) will be smaller compared to that experienced with a constant bit rate channel. It is also beneficial to other TCP connections which will experience rapid initial throughputs (during the TCP slow-start period) followed by convergence to a near constant value.

In practice, an Admission Control algorithm will be used to limit the number of users in the system and hence ensure that all accepted users are able to achieve their QoS demands. If insufficient resources are available to provide the throughput demands for all users then the Barrier Function approach gracefully reduces throughput to all users and maintains the throughputs of the Gold users higher than that of the Silver users. To illustrate this property, we increased the number of users by 25% to 15 Gold and 15 Silver users. We also increased the penalty coefficient $\beta$ by a factor of 5 for all users. The resulting throughputs are plotted in Figure 5. Note that both Gold and Silver users no longer achieve their desired throughput values but the variance of their throughput remains small and Gold users continue to maintain higher average throughputs than Silver users.

6. Conclusions

In this paper we proposed a method for providing QoS services over a shared high speed wireless data channel. The approach is based on using Barrier Functions to enforce QoS constraints while still allowing for user diversity gains. The scheduler is simple yet is flexible enough for most practical situations. We illustrated its usefulness by providing CSE bandwidths for two different class of users.

REFERENCES