DEPLOY CONTROL FOR CDF SCHEDULING USING MARKOV DECISION PROCESS

Phuong Bang C. Nguyen, Bhaskar D. Rao

Department of Electrical and Computer Engineering
University of California, San Diego

ABSTRACT

We consider the problem of controlling the user service delays for a system that employs a scheduling scheme based on Cumulative Density Functions (CDF) of user channels in a correlated Rayleigh fading environment. We first formulate and solve this control problem as a Markov Decision Process (MDP) on the system state space formed by the user channel conditions and delay times. The MDP formulation, however, has prohibitively high complexity for systems with typical number of users. We then propose an approximation to the MDP formulation that can achieve performance close to MDP optimal solution but has much lower complexity for large systems.

Index Terms— CDF, MDP, scheduling, QoS, delay

1. INTRODUCTION

In a wireless system, dynamic user scheduling is one of the most important techniques to maximize the system performance. One fundamental function of any scheduling policy in a multi-user system is to provide access fairness among the users while taking maximal advantage of multiuser diversity. Many scheduling schemes have been proposed over the years with many different performance and fairness criteria [1], [2]. In this paper, we investigate a particular scheduling scheme with a very rigorous notion of fairness, the CDF-based scheduling policy introduced in [3]. In CDF scheduling, the users are served when their channels are at their own best, independent of the discrepancies in channel probability distributions among the users.

While CDF-scheduling can guarantee fairness among all the users on a long-term basis when all the users have a chance to experience all different fades of their channels, there is no limit as to how long a particular user must wait to be served. If a user that has a long channel coherence time happens to be in a deep fade, the user’s service delay time can be very long, which can be unacceptable depending on the user’s applications. Service delay time is thus a very important Quality-of-Service (QoS) metric of any scheduling policy. In [4], the authors address this QoS issue using the effective bandwidth and capacity formulations. However, they do not consider the case where more than one users in the system have correlated channels, which is most likely the case in any real system. Other works in the literature that deal with service delay QoS such as [5] [6] do not address the CDF scheduling framework that we are interested in. In [7], the authors consider optimizing user queuing delays under a finite backlog scenario using Partially Observed MDP (POMDP). In contrast, our work is developed under the assumption of infinite backlog for all users, where the access fairness becomes critical. The work closest to ours in terms of problem formulation is [8], in which the authors also use MDP for solving optimal code allocation in packet scheduling for HSDPA systems. While our work starts out with the MDP framework, our MDP formulation, however, has completely different state and action spaces, and also a different optimization objective that seeks to maximize the CDF performance while providing service delay control. In addition, using the MDP formulation only as a guideline, we specifically design a low-complexity algorithm that works well for systems with large number of users.

2. SYSTEM MODEL

Let us consider a wireless system with $K$ users sharing the same wireless channel in a time division manner. Assume all the users experience independent Rayleigh fading with different coherence times due to differences in their mobility. Let $X_k, k = 1 \ldots K$, be the instantaneous SNR of user $k$. Let $U_k = F_{X_k}(X_k)$ be the CDF-transformed random variable for user $k$, where $F_{X_k}(x)$ is the Cumulative Distribution Function (CDF) for $X_k$. In CDF scheduling [3], the user is scheduled based on $U_k$ according to $k^* = \arg\max_k U_k^{(1/w_k)}$, where $k^*$ is the index of the selected user, $w_k$ is the time allocation fraction for user $k$, $\sum_{k=1}^{K} w_k = 1$.

We now divide the range of $U_k$, which is interval $[0, 1]$ on the y-axis on Fig 1, into $(M + 1)$ equal sized intervals with boundary values denoted by $u_i, i = 0, \ldots, (M + 1)$ (with $u_0 = 0, u_{M+1} = 1$). Let $i_k$ be the index of the quantized CDF-mapped value of the $X_k$, where $u_{i_k} \leq U_k < u_{i_k+1}$ as in Fig 1 (denoted as the CDF index henceforth). It can be seen that the CDF index $i_k$ captures the user’s relative channel condition via the corresponding quantized CDF value $u_{i_k}$.

This research was supported in part by the UCSD center for wireless communications and the NSF Grant CCF-1115645.
Consider a fading scenario where the fade is constant for a symbol duration with a CDF index \( i_k \) and the CDF index for the next symbol \( i_k' \) is either \( i_k, (i_k - 1) \) or \( (i_k + 1) \). For Rayleigh fading channels, the transition probability from one quantized SNR value to an adjacent one is derived in [9] as:

\[
Pr[i_k' = i + 1|i_k = i, i_k,i-1 = i_k] = \frac{N_i}{R_{t}(i)} = R_{t}(i) × p_i,
\]

where \( N_i = \sqrt{2\pi d_i \rho} f_m \exp(-\frac{d_i}{\rho}) \), \( R_{t}(i) = R_{t} × p_i \)

It is obvious that the channel SNR quantized according to the above CDF partitioning forms a discrete Markov process. When the scheduling interval \( T_{sched} \) is larger than one symbol duration, \( T_{sched}/T_{sym} = N > 1 \), let \( P \) be the 1-step transition matrix, we have an \( N \)-step transition:

\[
Pr[i_k' = j|i_k = i] = [P^N]_{i,j}, \text{ where } [P]_{i,j} = t_{i,j}.\]

Since the user channel conditions and, as we will see later, serving times as well can be modeled as discrete Markov processes, it is natural to consider the Markov Decision Process (MDP) for obtaining the optimal scheduling decisions.

### 3. MDP FORMULATION

In order to formulate an MDP, we need to define a system state vector \( s(n) \) at time \( n \), \( s(n) \in \mathcal{S} \), an action vector \( a(n) \in \mathcal{A} \), the transition probability \( Pr[s(n+1)|s(n), a(n)] \), and a reward function \( R(s(n), a(n)) \). The reward function \( R \) must be designed to reflect the objective of the scheduling policy. The optimal policy \( \pi^* \) is one that maximizes the expected total reward \( V^\pi(s) \) from any state \( s \) [10]:

\[
\pi^* = \arg\max_{\pi} V^\pi(s) = \arg\max_{\pi} \mathbb{E} \left[ \sum_{n=0}^{\infty} \gamma^n R(s(n), a(n))|s(0) = s \right]
\]

where \( \gamma < 1 \) is the discount factor. At time slot \( n \), we define the system state as follows:

\[
s(n) = [i_k(n), T_k(n)]_l, k = 1 \ldots K, \quad (2)
\]

where \( i_k(n) \) is the CDF index of the user \( k \) at time slot \( n \), \( T_k(n) = \min \{t_k(n), T_{k,th}\} \) with \( t_k(n) \) being the number of time slots since the last time user \( k \) was served, \( T_{k,th} \) a predefined maximum threshold used to keep the state space finite, which can potentially be assigned with a larger than normal weight to discourage staying in this state. \( T_{k}(n) \) forms a simple Markov process as shown in Fig 2.

The action vector \( a(n) \) is defined as one of the basis vector \( e_k \in \mathcal{B} \) of the vector space \( \mathbb{R}^K \)

\[
a(n) \in \mathcal{B} = \{[1, 0, \ldots, 0]^T, \ldots, [0, 0, \ldots, 1]^T\} \quad (3)
\]

The reward function \( R(n) \triangleq R(s(n)) \) is designed to both maximize the CDF of the chosen user and minimize the service time for all users.

\[
R(n) = [\hat{U}_{k,\pi}(n)]^{1/w_{k,\pi}} - \alpha(v, T(n)),
\]

where \( \hat{U}_{k,\pi}(n) = i_k, (n)/(M + 1) \) is the quantized CDF and \( w_{k,\pi} \) the time allocation for the selected user; \( \alpha \) is the delay penalty factor, \( v = [v_1, \ldots, v_K]^T \), the delay priority weighting, and \( T(n) = [T_1(n), \ldots, T_K(n)]^T \).

This reward function construction allows the MDP problem to simplify to one of maximizing the quantized CDF when \( \alpha = 0 \), which becomes CDF scheduling in [3] as \( M \to \infty \).
The transition probability can be obtained as follows:

\[ P_{s,s'}(a) = \Pr[s'|s, a] = \Pr[t'_k|t_k, \forall l \geq 2, t'_m, \forall m, t_j, \forall j, a] \]

\[ \times \Pr[t'_k|t'_m, \forall l > 2, t'_m, \forall m, t_j, \forall j, a] \]

\[ \times \Pr[t'_k|t'_m, \forall m > 2, t'_m, \forall m, t_j, \forall j, a] \times \Pr[t'_k|i_j, t_j, \forall j, a] \]

As \( i_k(n) \)'s are independent across \( k \) due to independent fading of the users, and the user's service time depends only its previous state and the system's action, we have:

\[ \Pr[t'_k|t_k, a] = \begin{cases} 1 & \text{if } (k \neq k^*, t'_k = \min \{t_k + 1, T_k, t_h\}) \\ 0 & \text{otherwise} \end{cases} \]

The optimal MDP policy is obtained from the following Value Iteration recursion, which converges as \( l \to \infty \):

\[ V_l(s) = \max_{a \in A} \left[ R(s, a) + \gamma \sum_{s' \in S} P_{s,s'}(a)V_{l-1}(s') \right] \]

where \( V^*(s) \) satisfies the Bellman's Optimality Equation

\[ V^*(s) = \max_{a \in A} \left[ R(s, a) + \gamma \sum_{s' \in S} P_{s,s'}(a)V^*(s') \right] \]

By ignoring the second term in (5), which corresponds to a small value of \( \gamma \), we obtain an approximate policy \( \tilde{\pi}(s) \)

\[ \tilde{\pi}(s) = \arg\max_{1 \leq a \leq K} \left[ R(s, a) + \gamma \sum_{s' \in S} P_{s,s'}(a)R(s', a) \right] \]

where \( R_i(a) \) is the immediate reward and \( R_f(a) \) the expected future reward if user \( a \) is currently selected. Note that in (6) we use the user index \( a \) instead of the action vector \( a \) as the action vector \( a \) defined in (3) has only one nonzero component at some location \( a \). Policy \( \tilde{\pi}(s) \) in (6) still suffers from a large number of states. To avoid this problem, we approximate the future reward portion \( R_f \) with an estimate as follows:

\[ R_f(a) \approx \max_k \sum_{i_k=0}^M P[i_k'|i_k] \tilde{R}(k, a), \]

where \( \tilde{R}_k(a) \) is the expected reward if user \( k \) is selected in the next step; \( \tilde{R}(k, a) = \tilde{R}(s', a) \) is the reward obtained if user \( k \) is selected at the next step given that user \( a \) is selected in the current step. Here we have:

\[ s' = [t', T'(k, a)]^T, \quad t' = [i'_1, i'_2, \ldots, i'_K]^T \]

\[ T'(k, a) = [t'_1|k,a, t'_2|k,a, \ldots, t'_K|k,a] \]

\[ t'_j|k,a = \begin{cases} 0 & \text{when } j = k \\ t_j + 1 & \text{otherwise} \end{cases} \]

\[ R_i(a) = R(s, a) = [\tilde{U}_k]^{(1/\nu_a)} - \alpha(\bar{U}_a, T(a)) \]

\[ \tilde{R}(k, a) = [\tilde{U}_k]^{(1/\nu_k)} - \alpha(\bar{U}_a, T'(k, a)) \]

\[ U_k = i_a/(M + 1), \quad \bar{U}_k = i_k/(M + 1) \]

In (7), the future reward is the maximum of all users' expected rewards, which is a reasonable estimate since the future user selection maximizes this reward. Finally, we obtain the following policy:

\[ a^* = \arg\max_{1 \leq a \leq K} \left[ R(s, a) + \gamma \sum_{i_k=0}^M P[i_k'|i_k] \tilde{R}(k, a) \right] \]
In order to compare the AMDP policy and the MDP policy, we adjust the parameters of AMDP to achieve the same SNR performance as MDP. We also plot the performance of the Round-Robin (RR) and the Proportional Fairness (PF) schemes for comparison. The parameters of the Proportional Fairness Scheme are adjusted such that its delays are in the same neighborhood of those in the AMDP scheme.

It can be seen from figure 3 that the unconstrained CDF scheme achieves the best SNR performance. However, figure 4 shows that CDF-policy can have very large delay times (more than 200 slots in this case). The MDP takes a performance hit to in order to lower delay time. AMDP can achieve the same performance as MDP at a slightly worse delay performance, which is still much better than that of the CDF policy. The PF scheme behaves worse than MDP/AMDP schemes in both delay and SNR performance. Both MDP/AMDP policies can drastically reduce the maximum delays as well as the delay variances.

In order to examine the performance of the AMDP policy in larger systems, we simulate a system of $K = 10$ users and $M = 63$ with user speeds ranging from $v_1 = 2$ m/s to $v_{10} = 30$ m/s. This system is already too large for the exact MDP policy as discussed in section 4. In Figure 5, we plot the delay performance for three users 1, 5, and 10, with lowest, middle, and highest mobility. As expected, Figure 5 shows that the AMDP policy significantly reduces the service delays for all users (less than 150 slots) compared to the CDF policy (more than 10,000 slots in some cases). Figure 6 shows the small performance loss incurred by AMDP compared to the unconstrained CDF policy. The PF scheme with similar delays as the AMDP policy behaves a lot worse than AMDP.

### 6. CONCLUSIONS

In this paper, we set up an MDP problem specifically for the CDF-scheduling framework. We then demonstrate through simulations that it can drastically reduce the user’s delay time at a small cost to the throughput performance. We also derive an Approximate MDP algorithm that can achieve similar delay control with much lower complexity. Our future work will include detailed mathematical analysis of the performance bounds afforded by this low complexity policy.
7. REFERENCES


