DELAY CONTROL FOR CDF SCHEDULING USING MARKOV DECISION PROCESS

PhuongBang 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.

This research was supported in part by the UCSD center for wireless communications and the NSF Grant CCF-1115645.

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 $X_k, k = 1 \dots K$, 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 $k^* = \underset{k}{\operatorname{argmax}}\, U_k^{(1/w_k)}$, 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 $u_i, i = 0, \dots, (M + 1)$ (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}.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

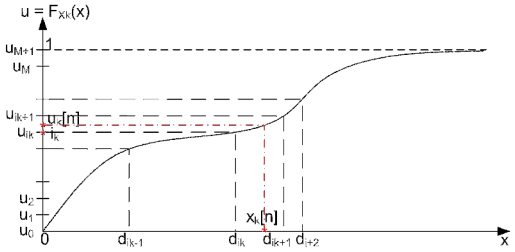

Fig. 1. CDF Partitioning

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:

$$t_{i,i+1} \approx \frac{N_{i+1}}{R_t^{(i)}} \text{ and } t_{i,i-1} \approx \frac{N_i}{R_t^{(i)}}, \tag{1}$$

where $t_{i,i+1} \triangleq \Pr[i'_k = i+1 | i_k = i]$, $t_{i,i-1} \triangleq \Pr[i'_k = i-1 | i_k = i]$, N_{i+1} is the average rate of the instantaneous SNR crossing the threshold X_i, N_i is the average rate of the instantaneous SNR crossing the threshold X_{i-1}, $R_t^{(i)}$ is the average rate transmitted when the instantaneous SNR is in the i^{th} interval. Also according to [9], we have

$$N_i = \sqrt{\frac{2\pi d_i}{\rho}} f_m \exp\left(-\frac{d_i}{\rho}\right), \; R_t^{(i)} = R_t \times p_i$$

where ρ is the mean of the squared Rayleigh fading amplitude, d_i is the quantization boundary in Fig. 1, $f_m = f_c v/c$ is the max Doppler shift, with v being the user velocity, f_c the carrier frequency, and c the speed of light. R_t is the symbol rate, and $p_i = \Pr[i_k = i]$. Since the quantization is uniform in the CDF domain per our construction, we conclude that $p_i = \Pr[i_k = i] = 1/(M+1)$. As all the users share the same RF channel, the symbol rate $R_t = 1/T_{sym}$ is the same for all users, where T_{sym} is the symbol interval, which depends only on the bandwidth of the channel. Thus the following result is true for all users:

$$R_t^{(i)} = R_t \times p_i = \frac{1}{(M+1)T_{sym}}$$

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 $\mathbf{P}$ be the 1-step transition matrix, we have an N-step transition:

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

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.

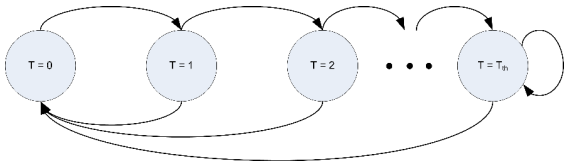

Fig. 2. Markov Process For Serving Times

3. MDP FORMULATION

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

$$\begin{aligned}\pi^* &= \underset{\pi}{\text{argmax}}\, V^\pi(\mathbf{s}) \\ &= \underset{\pi}{\text{argmax}}\, \mathbf{E}\left[\sum_{n=0}^{\infty} \gamma^n R(\mathbf{s}(n), \mathbf{a}(n)) | \mathbf{s}(0) = \mathbf{s}\right]\end{aligned}$$

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

$$\mathbf{s}(n) = [i_k(n), T_k(n)], k = 1 \ldots K, \tag{2}$$

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

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

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

$$R(n) = [\hat{U}_{k^*}(n)]^{1/w_{k^*}} - \alpha \langle \underline{\nu}, \mathbf{T}(n) \rangle,$$

where $\hat{U}_{k^*}(n) = i_{k^*}(n)/(M+1)$ is the *quantized* CDF and w_{k^*} the time allocation for the selected user; α is the *delay penalty factor*, $\underline{\nu} = [\nu_1, \ldots, \nu_K]^T$, the *delay priority weighting*, and $\mathbf{T}(n) = [T_1(n), \ldots, T_K(n)]^T$.

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 $M \to$

∞. The transition probability can be obtained as follows:

$$\begin{aligned}
&\mathrm{P}_{\mathbf{s},\mathbf{s}'}(\mathbf{a}) = \Pr[\mathbf{s}'|\mathbf{s},\mathbf{a}] \\
&= \Pr[i'_1|i'_l, \forall l \geq 2, t'_m, \forall m, i_j, t_j, \forall j, \mathbf{a}] \\
&\times \Pr[i'_2|i'_l, \forall l \geq 3, t'_m, \forall m, i_j, t_j, \forall j, \mathbf{a}] \times \dots \\
&\times \Pr[i'_K|t'_m, \forall m, i_j, t_j, \forall j, \mathbf{a}] \times \\
&\times \Pr[t'_1|t'_m, \forall m \geq 2, i_j, t_j, \forall j, \mathbf{a}] \times \Pr[t'_K|i_j, t_j, \forall j, \mathbf{a}]
\end{aligned}$$

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:

$$\begin{aligned}
&\Pr[i'_k|i'_l, \forall l > k, i_j, t_j, \forall j, \mathbf{a}] = \Pr[i'_k|i_k] \\
&\Pr[t'_k|t'_m, \forall m > k, i_j, t_j, \forall j, \mathbf{a}] = \Pr[t'_k|t_k, \mathbf{a}] \\
&\Rightarrow \mathrm{P}_{\mathbf{s},\mathbf{s}'}(\mathbf{a}) = \prod_{k=1}^{K} \Pr[i'_k|i_k] \prod_{k=1}^{K} \Pr[t'_k|t_k, \mathbf{a}]
\end{aligned}$$

Let $\mathbf{s} = [i_1, i_2, \dots, i_K; t_1, t_2, \dots, t_{k^*}, \dots t_K]^T$ and $\mathbf{a} = \mathbf{e}_{k^*}$. That is, user k^* is chosen to be served.

$$\Pr[t'_k|t_k, \mathbf{a}] = \begin{cases} 1 & \text{if } (k \neq k^*, t'_k = \min\{t_k + 1, T_{k,th}\}) \\ & \quad \text{or } (k = k^*, t'_k = 0) \\ 0 & \text{otherwise} \end{cases}$$

In the next time slot, the state vector becomes $\mathbf{s}^+ = [i'_1, i'_2, \dots, i'_K; t_1 + 1, t_2 + 1, \dots, t'_{k^*} = 0, \dots, t_K + 1]^T$.

$$\Rightarrow \mathrm{P}_{\mathbf{s},\mathbf{s}'}(\mathbf{a}) = \begin{cases} \prod_{k=1}^{K} \Pr[i'_k|i_k] & \text{if } \mathbf{s}' = \mathbf{s}^+ \\ 0 & \text{otherwise} \end{cases}$$

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

$$V_l(\mathbf{s}) = \max_{\mathbf{a} \in \mathcal{A}} [R(\mathbf{s}) + \gamma \sum_{\mathbf{s}' \in \mathcal{S}} P_{\mathbf{s},\mathbf{s}'}(\mathbf{a}) V_{l-1}(\mathbf{s}')] \qquad (4)$$

4. LOW COMPLEXITY SOLUTION

The definition of the state space $\mathcal{S}$ according to (2) could lead to an extremely large number of states:

$$|\mathcal{S}| = (M+1)^K (T_{th}+1)^K = [(M+1)(T_{th}+1)]^K$$

For instance, for a system with 10 users ($K = 10$), let the quantization size to be $M = 4$, and the threshold be twice the number of users $T_{th} = 20$, the number of states is then

$$|\mathcal{S}| = (5 \times 21)^{10} \approx 10^{20} \text{ !!!}$$

Thus, the exact MDP approach is not feasible for practical systems with typical number of users. To find a low complexity solution, we start from the following relation for the optimal MDP policy $\pi^*(s)$:

$$\pi^*(s) = \operatorname*{argmax}_{\mathbf{a} \in \mathcal{A}} [R(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}' \in \mathcal{S}} P_{\mathbf{s},\mathbf{s}'}(\mathbf{a}) V^*(\mathbf{s}')]$$

where $V^*(\mathbf{s})$ satisfies the *Bellman's Optimality Equation*

$$V^*(s) = \max_{\mathbf{a} \in \mathcal{A}} [R(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}' \in \mathcal{S}} P_{\mathbf{s},\mathbf{s}'}(\mathbf{a}) V^*(\mathbf{s}')] \qquad (5)$$

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

$$\tilde{\pi}(s) = \operatorname*{argmax}_{1 \leq a \leq K} [\underbrace{R(\mathbf{s}, a)}_{R_i(a)} + \gamma \underbrace{\sum_{\mathbf{s}' \in \mathcal{S}} P_{\mathbf{s},\mathbf{s}'}(a) R(\mathbf{s}', a)}_{R_f(a)}], \qquad (6)$$

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 $\mathbf{a}$ as the action vector $\mathbf{a}$ 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:

$$R_f(a) \approx \max_k \underbrace{\sum_{i'_k=0}^{M} P[i'_k|i_k] \hat{R}(k, a)}_{\bar{R}_k(a)}, \qquad (7)$$

where $\bar{R}_k(a)$ is the expected reward if user k is selected in the next step; $\hat{R}(k, a) = R(\mathbf{s}', a)$ 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:

$$\begin{aligned}
&\mathbf{s}' = [\mathbf{i}', \mathbf{T}'(k, a)]^T, \ \mathbf{i}' = [i'_1, i'_2, \dots, i'_K]^T \\
&\mathbf{T}'(k, a) = [t'_{1|k,a}, t'_{2|k,a}, \dots, t'_{K|k,a}] \\
&t'_{j|k,a} = \begin{cases} 0 & \text{when } j = k \\ 1 & \text{when } j = a \\ t_j + 1 & \text{otherwise} \end{cases} \\
&R_i(a) = R(\mathbf{s}, a) = [\hat{U}_a]^{(1/w_a)} - \alpha \langle \underline{\nu}, \mathbf{T}(a) \rangle \\
&\hat{R}(k, a) = [\hat{U}'_k]^{(1/w_k)} - \alpha \langle \underline{\nu}, \mathbf{T}'(k, a) \rangle \\
&\hat{U}_a = i_a/(M+1), \ \hat{U}'_k = i'_k/(M+1)
\end{aligned}$$

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:

Table 1. Approximate MDP Policy (AMDP)

$$a^* = \operatorname*{argmax}_{1 \leq a \leq K} [R(\mathbf{s}, a) + \gamma \max_k \sum_{i'_k=0}^{M} P[i'_k|i_k] \hat{R}(k, a)] \qquad (8)$$

In the AMDP policy (8), the delay penalty factor α controls the delay-induced penalty. Similar to the exact MDP

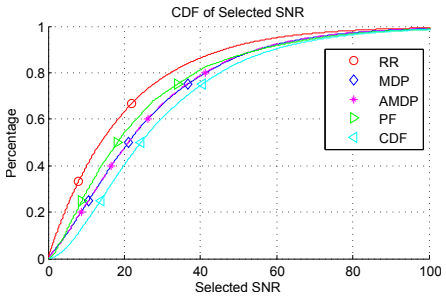

Fig. 3. System Selected SNR: CDF gets the best SNR, AMDP can achieve MDP's SNR with slightly higher delays.

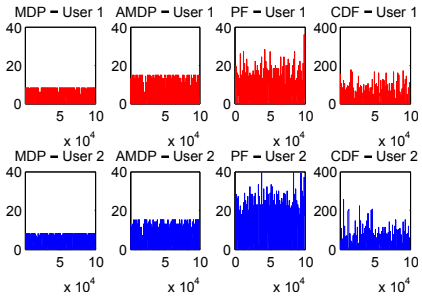

Fig. 4. Serving Time Delays: CDF has worst delays, MDP delays are lowest, AMDP delays are slightly worse.

problem in section 3, this problem simplifies to CDF scheduling in [3] when $\alpha = 0$ and $M \to \infty$. The discount factor γ allows the effects of the user coherence time to be used to further maximize the reward.

5. NUMERICAL RESULTS

We simulate a small system with parameters shown in table 2. 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.

Table 2. Simulation Parameters

Parameter	Value	
Number of users	2	K
Quantization size	7	M
Carrier frequency	1 GHz	f_c
Channel bandwidth	15 KHz	
User speeds	$v_1 = 2, v_2 = 25$	m/s
Average channel SNR	30	ρ
Serving time threshold	6	T_{th}
Time allocation fraction	$w_1 = 0.5, w_2 = 0.5$	
Delay priority weighting	$\nu_1 = 0.5, \nu_2 = 0.5$	
MDP discount factor	0.9	γ
Delay penalty factor	0.4	α

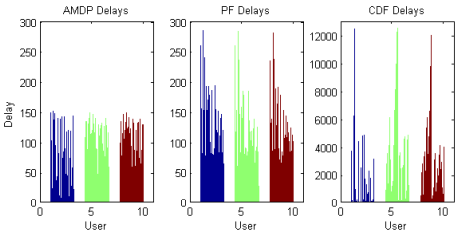

Fig. 5. Delay Comparisons for a 10-User System: AMDP significantly reduces the delays compared to CDF.

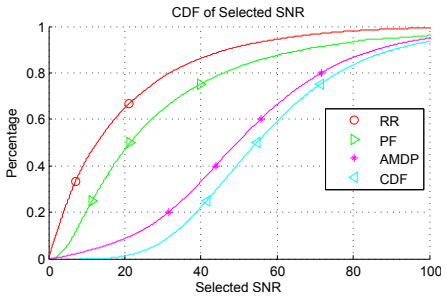

Fig. 6. Performance Comparisons for a 10-User System: AMDP takes a small loss, but is much better than PF and RR.

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

[1] Matthew Andrews, "A survey of scheduling theory in wireless data networks," in *Wireless Communications*, Prathima Agrawal, PhilipJ. Fleming, Lisa Zhang, DanielMatthew Andrews, and George Yin, Eds., vol. 143 of *The IMA Volumes in Mathematics and its Applications*, pp. 1–17. Springer New York, 2007.

[2] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–18, 2013.

[3] Daeyoung Park, Hanbyul Seo, Hojoong Kwon, and Byeong Gi Lee, "Wireless packet scheduling based on the cumulative distribution function of user transmission rates," *Communications, IEEE Transactions on*, vol. 53, no. 11, pp. 1919–1929, 2005.

[4] Daeyoung Park and Byeong Gi Lee, "QoS support by using cdf-based wireless packet scheduling in fading channels," *IEEE Transactions on Communications*, vol. 54, no. 11, pp. 2051 – 2061, Nov 2006.

[5] M.J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1728–1736.

[6] Ramtin Kazemi Beidokhti, Mohammad Hossein Yaghmaee Moghaddam, and Jalil Chitizadeh, "Adaptive QoS scheduling in wireless cellular networks," *Wireless Networks*, vol. 17, no. 3, pp. 701–716, Apr. 2011.

[7] Huang Huang and V.K.N. Lau, "Delay-optimal user scheduling and inter-cell interference management in cellular network via distributive stochastic learning," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 12, pp. 3790–3797, December 2010.

[8] H. Al-Zubaidy, I. Lambadaris, and J. Talim, "Optimal scheduling in high-speed downlink packet access networks," *ACM Transactions on Modeling and Computer Simulation*, vol. 21, pp. 3:1–3:27, Dec 2010.

[9] H. S. Wang and N. Moayeri, "Finite-state markov channel - a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 163 – 171, Feb 1995.

[10] Stuart Russell and Norvig Peter, "Artificial intelligence, a modern approach," pp. 645–656. Prentice Hall, 2010.