An On-Line Learning Algorithm for Energy Efficient Delay Constrained Scheduling over a Fading Channel

Nitin Salodkar, Abhijeet Bhorkar, Abhay Karandikar, *Member, IEEE,* and Vivek S. Borkar, *Fellow, IEEE*

***Abstract*—In this paper, we consider the problem of energy efficient scheduling under average delay constraint for a single user fading channel. We propose a new approach for on-line implementation of the optimal packet scheduling algorithm. This approach is based on reformulating the value iteration equation by introducing a virtual state called *post-decision state*. The resultant value iteration equation becomes amenable to on-line implementation based on stochastic approximation. This approach has an advantage that an explicit knowledge of the probability distribution of the channel state as well as the arrivals is not required for the implementation. We prove that the on-line algorithm indeed converges to the optimal policy.**

***Index Terms*—Energy Efficient Scheduling, Stochastic Approximation, Constrained Markov Decision Processes**

I. Introduction

Over the past few years, wireless networks have witnessed remarkable growth both in terms of number of subscribers and number of applications. While the earlier growth in wireless networks was driven primarily by cellular telephony, the recent years have seen an increase of wireless data and multimedia applications [1]. These applications call for efficient resource allocation. The traditional view of resource allocation in packet networks is to treat the network architecture as a *layered architecture* [2]. In this layered architecture model, the network and link layers deal with packet level quality of service attributes like throughput, packet delay, loss etc., while the physical layer deals with information theoretic limits for reliable communication and practical means of achieving it.

Wireless channel is characterized by *fading*, i.e., signal strength variations over time, frequency and space due to constructive and destructive interference caused by multipaths. Moreover, limited battery life at mobile hosts poses an additional challenge in system design and resource allocation. Recent studies suggest that significant performance gains can be obtained by *cross layer* approach [3] where the physical layer characteristics can be exploited for resource allocation at the network layer. See [4] for an excellent overview of cross layer resource allocation and other related issues.

There has also been a lot of interest in studying information theoretic capacity aspects of fading channels. A comprehensive review of recent developments has been given in [5]. For a single user fading channel, it has been shown [6] that if the transmitter is aware of the channel state information, then the channel capacity under the constraint on average power can be maximized by the 'water-filling' power allocation over the fading states. This suggests that we should transmit more information in good channel states and less in bad channel states in order to maximize the long term average throughput. (See also [7] for other notions of capacity of fading channel.)

Apart from fading characteristics, energy efficiency is also an important concern for wireless devices. It can be argued [8] that for most wireless communication systems, the power required to transmit a bit 'reliably' is an increasing and strictly convex function of the transmission rate for a given fading channel state. Thus, if we transmit at a lower rate, we can conserve energy, but this leads to packet delay at the network layer. Moreover, as discussed above, since capacity is maximized by transmitting in good channel states, the transmitter should defer the transmission of packets if the channel is bad. However, this also contributes to packet delay at the network layer. This indicates a fundamental trade-off between the average power and the average delay.

In practice, it can be assumed that the channel coherence times are reasonably long so that 'near' capacity rates can be achieved by employing practically implementable codewords and yet the coherence times are smaller than the 'packet delay time-scale' of interest at the network layer. Accordingly, in this paper, we consider the problem of scheduling packets for a single user over a point to point link where the objective is to minimize the average power (required for 'reliably' transmitting at a given rate) subject to a constraint on the average (queueing) delay.

A. Related Work

The problem of energy efficient scheduling with average delay constraint was first addressed in the pioneering work of [8], [9]. Subsequently, the other works [10], [11], [12], [13], [14], [15] have also considered this problem under various

Nitin Salodkar is with the Department of Computer Science and Engineering, IIT Bombay, Powai, Mumbai 400076, India. (Email: nitins@cse.iitb.ac.in). Salodkar's work supported in part by an Infosys Scholarship.

Abhijeet Bhorkar is with the Department of Electrical Engineering, University of California, San Diego, 9500 Gilman Dr., La Jolla, CA 92093, USA. (Email: abhorkar@ucsd.edu).

Abhay Karandikar is with the Department of Electrical Engineering, IIT Bombay, Powai, Mumbai 400076, India. (Email: karandi@ee.iitb.ac.in). Karandikar's work supported by Department of Science and Technology (DST grant 05DS025).

Vivek S. Borkar is with the School of Technology and Computer Science, Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai 400005, India. (Email: borkar@tifr.res.in). Borkar's work supported in part by grant no. 2900-IT-1 from the *Centre Franco-Indien pour la Promotion de la Recherche Avancee.*

assumptions on packet arrival and channel state processes. Interestingly, this problem has also been considered for non-fading channel [16] and also for the case when the power required for transmission is a linear function of rate [17]. The model considered in this paper is similar to that of [8], [10], [11], [12], [13], [14], [15] with a single user fading channel. Under this model, the scheduling policy can be considered as a control policy which decides the number of packets to be transmitted to maximize an average cost subject to some constraint. Thus this problem falls within the framework of constrained Markov decision process [18].

In [8], [9], the trade-off between the average delay and the average power has been analyzed. The delay-power trade-off has also been quantified in the region of asymptotically large delays. In [9], structural results for optimal policy have been derived. Structural results have also been discussed in [11] for a policy which minimizes the average delay subject to a constraint on the average power. It is proved in [9] and [11] that there exists an optimal stationary policy which increases as the buffer occupancy increases, and decreases as the channel state goes from good to bad. What this means in physical terms is that for a fixed channel gain, the greater the queue length, the more you transmit, and for a fixed queue length, the better the channel, the more you transmit. A characterization of the optimal scheduler has also been provided in [12] in terms of a smaller class of deterministic schedulers.

While the existence of a stationary optimal policy for the average cost problem has been considered in [11] when the packet arrival process is independent and identically distributed (i.i.d.), the problem becomes much more difficult when the arrival process is Markovian. For this case, in [8], only the unconstrained average cost dynamic programming formulation has been given. Subsequently, the model of [8] and [11] has been extended in [14] where the authors consider a more general state space version of the average cost Constrained Markov Decision Problem (CMDP). In this model, both the arrival and the channel state process are considered to be Markovian. In [13], a discrete state space version of the same problem has been considered for correlated arrivals and correlated fading. In [10], it is observed that while structural results of an optimal policy can be derived, it is very difficult to compute the optimal policy due to large state space size. Accordingly, the authors suggest a suboptimal policy that is of ‘thresholding’ type. They also suggest a mechanism to derive the optimum thresholds for queue length and channel gain.

While all the above approaches have provided significant insights into the problem, none of them explicitly deals with the *computation* of optimal packet scheduling algorithm. Consequently, the practical implementation of the policy remains an important open issue. This is primarily due to the following two reasons-

1) Since the state space is very large, the standard dynamic programming algorithms are very hard to implement. The preferred technique for CMDP has been linear programming [18]. Combined with the more recent approach based on function approximation [19], this holds great promise. The structural results of the policy may help in the choice of basis functions in function approximation based computation. But none of the related work discussed above seems to have explored this issue in developing implementable optimal packet scheduling algorithm.
2) Secondly and more importantly, the computation of the optimal policy using the above mentioned techniques assumes the knowledge of the underlying model. This means that the knowledge of the probability distribution of both the arrival process and the channel state is necessary for the computation of optimal policy. This is usually not the case in practice. In [10], the authors have proposed a suboptimal algorithm but even here, the computation of the appropriate thresholds seems to assume the knowledge of the arrival process and channel state distribution.

B. Our Contributions

The primary contribution of this paper is to propose an on-line learning algorithm to compute the optimal packet scheduling policy for single user fading channel, where the model is not known. Recall that the Relative Value Iteration Algorithm (RVIA) [20] is a well known algorithm for determining an optimal policy for classical MDP. In this paper, we propose a new approach for on-line implementation of RVIA for the average cost CMDP such as the one studied in this paper. This approach is based on reformulating the value iteration equation by introducing a virtual state called *post-decision state*. The resultant value iteration equation has a nice structure that lends itself naturally to its on-line implementation based on stochastic approximation framework. Note that like all other works [8], [10], [11], [13], [14], [15], we assume that the transmitter is aware of the channel state information at the beginning of each time slot. But unlike others, an explicit knowledge of the probability distribution of the channel state as well as the arrival process is not required for the proposed implementation. We also prove that the proposed on-line algorithm indeed converges to the optimal policy. The challenge in developing an on-line algorithm has also been recognized recently in [21] where the author has extended the formulation of [8] for multi-user setting and proposed an algorithm that does not require the model knowledge. However, this algorithm is only order-optimal.

The rest of the paper is organized as follows. We begin this paper by discussing the system model in Section II. The formulation of the problem within the CMDP framework is presented in Section III. This parallels the development in [8] and other related work discussed above. Our point of departure will be the development after the formulation of the problem within the framework of CMDP. In Section IV, we introduce the concept of a post-decision state and reformulate the value iteration. In Section V, we propose an on-line algorithm based on the post-decision state and stochastic approximation. Convergence of the on-line algorithm to the optimal policy is proved in the Appendix. For illustration purposes, we implement the algorithm and demonstrate its performance in a wireless system in Section VI. We conclude the paper with directions for future research in Section VII.

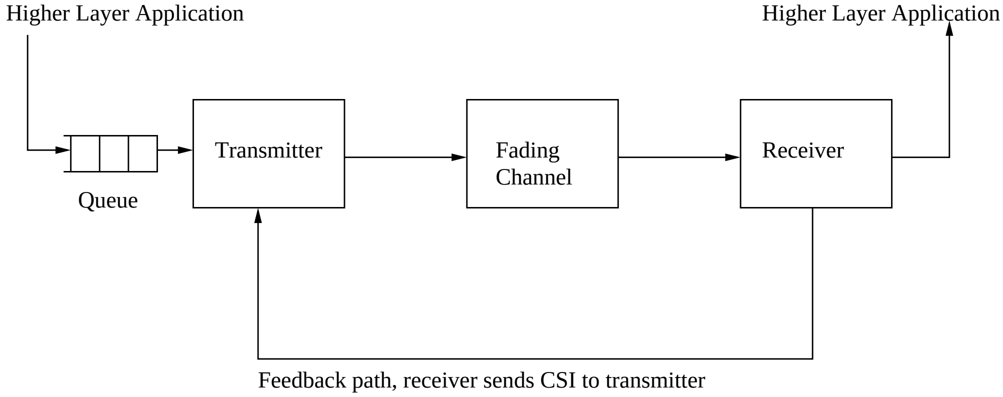

Fig. 1. System model

II. System Model

In this section, we discuss the system model. As illustrated in Figure 1, we consider a single user fading wireless channel. Time is divided into slots of equal duration which is normalized to unity. Packets arrive at the transmitter buffer and get queued until they are transmitted. Let $A_n \in \mathcal{A} \triangleq \{0, \ldots, A\}$ denote the number of arrivals just at the beginning of slot n. The packet arrival process $\{A_n\}$ is assumed to be an i.i.d. sequence. For simplicity, we make an additional assumption that every packet has a constant size, say ℓ bits. Let Q_n denote the queue length at the beginning of slot n and U_n denote the number of packets transmitted in slot n, then the dynamics of queue length can be expressed as

$$Q_{n+1} = \max\{Q_n - U_n, 0\} + A_{n+1}. \quad (1)$$

In practice, the transmitter buffer may have a finite size, say B and therefore $Q_n \in \mathcal{Q} \triangleq \{0, \ldots, B\}$. However, compared to the packet arrival rate, we assume B to be large enough so as to neglect the probability of buffer overflow and hence packet drops. Thus we do not consider the effect of finite buffer size in the queue dynamics mentioned above.

The channel is assumed to be time varying with *block fading* model, i.e., channel is constant over a slot duration and changes only at slot boundaries. Under this model, if the user transmits a signal y_n in slot n, then the received signal R_n is given by

$$R_n = H_n y_n + W_n, \quad (2)$$

where H_n[1] denotes the complex channel gain due to fading and W_n denotes the complex additive white Gaussian noise with zero mean and variance N_0[2]. Let $X_n = |H_n|^2$ be called the channel state at time n and W denote the received signal bandwidth. If the transmitted signal corresponds to transmitting $U_n = u$ packets, each of ℓ bits, then following the discussion in [8], the power required for error-free reliable communication when $X_n = x$ is given by,

$$P(x, u) = \frac{WN_0}{x}\left(2^{\frac{u\ell}{W}} - 1\right). \quad (3)$$

Note that for a given x, the transmission power $P(x, u)$ is an increasing and strictly convex function of u. Though the complex channel gain H_n (thereby X_n) may be a continuous random variable, we make a simplifying assumption that X_n takes values only from a finite state space $\mathcal{X}$ and varies from slot to slot in an i.i.d. fashion. This approximation is usually justified in practice and also has been used by other authors [8], [10].

The state of the above system S_n at time n can be described by the two tuple, $S_n = (Q_n, X_n)$, comprising of the queue length and the channel state. Note that the state space of the system $\mathcal{S} = \mathcal{Q} \times \mathcal{X}$ is discrete and finite. $\{U_n\}$ denotes the control process taking values from the finite action space $\mathcal{U} = \{0, \ldots, B\}$. The control action determines the number of packets U_n to be transmitted in a slot n. Since the transmitter can at most transmit all the packets that are present in the buffer at any time slot, we have $U_n \leq Q_n$. The control or scheduling policy is a sequence of functions $\{\mu_n\}$ where μ_n specifies U_n (more generally, the conditional law thereof) given the past history of the system state and past controls up to time n. The conditional law of the state S_{n+1} of the underlying dynamical system given this history depends only upon the state S_n and the control μ_n, thus making it a Markov decision process. Let $p : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to [0, 1]$ denote the state transition probability of this Markov Decision Process.

We note here that we can also handle Markov arrivals by augmenting the state space further to include the arrival process. The analysis will be similar. We avoid this generality in the interest of notational simplicity.

III. Problem Description

As discussed above, for a given x, the power required to transmit u packets, i.e., $P(x, u)$ is a convex function of u. Thus from an energy efficiency perspective, the policy should be to transmit packets in small chunks. This may, however, lead to queueing of packets thereby causing delay. Our objective is to minimize the average transmission power subject to constraint on the average packet delay.

[1] Random variables (RVs) are denoted with capital letters while values taken by the RVs are denoted by small letters.

[2] Usually, H_n is also assumed to be zero mean complex Gaussian random variable with unit variance under Rayleigh fading model. As demonstrated later, the knowledge of explicit probability distribution of the channel gain is not required for our approach.

A. Formulation as a Constrained Markov Decision Problem (CMDP)

By Little's law [2], the average delay $\bar{D}$ is related to the average queue length $\bar{Q}$ as,

$$\bar{Q} = \bar{a}\bar{D}, \tag{4}$$

where $\bar{a}$ is the average arrival rate. In the rest of the paper, we treat average delay as synonymous with average queue length and ignore the proportionality constant $\bar{a}$. Let $c_p(S_n, U_n)$ denote the cost in terms of power required in transmitting U_n packets when the state is S_n. Thus $c_p(S_n, U_n) = P(X_n, U_n)$. Since the packets get queued, they suffer a cost of buffering. Let $c_q(S_n, U_n) \triangleq Q_n$. The time averaged power and queue length can be expressed as,

$$\begin{aligned} \bar{P} &= \limsup_{N\to\infty} \frac{1}{N}\mathbf{E}\sum_{n=1}^{N} c_p(S_n, U_n), \quad \text{and} \\ \bar{Q} &= \limsup_{N\to\infty} \frac{1}{N}\mathbf{E}\sum_{n=1}^{N} c_q(S_n, U_n), \end{aligned} \tag{5}$$

respectively. Our objective is to design a scheduler that minimizes $\bar{P}$ subject to a constraint (say, δ) on $\bar{Q}$. It is a CMDP with average cost and finite state and action spaces. It is well known that a stationary randomized optimal policy exists [18]. Hence we concentrate only on stationary randomized policies characterized by $\mu(\cdot|s) : s \in \mathcal{S} \to$ probability measures on $\mathcal{U}$. That is, $\mu(\cdot|s)$ for each state s specifies the distribution with which the control in that state is applied. We assume irreducibility of the chain under such policies. Then $\{S_n\}$ is an ergodic Markov chain and thus has a unique stationary distribution ρ^μ. Let $\mathbf{E}^\mu$ denote the expectation w.r.t. ρ^μ. Under a randomized policy μ, the costs in (5) can be expressed as,

$$\bar{P}^\mu \triangleq \mathbf{E}^\mu\Big[c_p(S_n, \mu(S_n))\Big] = \sum_{u,s} \rho^\mu(s)\mu(u|s)c_p(s, \mu(s)), \tag{6}$$

and,

$$\bar{Q}^\mu \triangleq \mathbf{E}^\mu\Big[c_q(S_n, \mu(S_n))\Big] = \sum_{u,s} \rho^\mu(s)\mu(u|s)c_q(s, \mu(s)), \tag{7}$$

respectively. Then the scheduler objective can be stated as:

$$\text{Minimize } \ \bar{P}^\mu \ \text{ subject to } \ \bar{Q}^\mu \leq \delta. \tag{8}$$

B. The Lagrangian Approach

The constrained problem in (8) can be converted into an unconstrained one using standard Lagrangian approach [18], [22]. In this section, we discuss the Lagrangian formulation and the corresponding dynamic programming equation. The treatment in this section is standard and follows [20], [22]. Nevertheless, we include it here for introducing the notation and for an easier understanding of the rest of the paper.

Let $\lambda \geq 0$ be a real number. Define $c : \mathcal{R}^+ \times \mathcal{S} \times \mathcal{U} \to \mathcal{R}$ as follows,

$$c(\lambda, s, u) = c_p(s, u) + \lambda(c_q(s, u) - \delta). \tag{9}$$

Note that the function $c(\cdot, \cdot, u)$ is a strictly convex function of u (as the power required to transmit u packets is a strictly convex function of u). The unconstrained problem is to determine an optimal stationary policy $\mu^*(\cdot)$ that minimizes

$$L(\mu, \lambda) = \mathbf{E}^\mu\Big[c(\lambda, S_n, \mu(S_n))\Big], \tag{10}$$

for a particular value of λ called the Lagrange Multiplier (LM). $L(\cdot, \cdot)$ is called the Lagrangian. The following dynamic programming equation provides the necessary condition for optimality of the policy.

$$V(s) = \min_u \Big[c(\lambda, s, u) - \beta + \sum_{s'} p(s, u, s')V(s')\Big], \quad s' \in \mathcal{S}, \tag{11}$$

where $\beta \in \mathcal{R}$ is uniquely characterized as the corresponding optimal cost per stage. If we impose $V(s^0) = 0$ for a fixed $s^0 \in \mathcal{S}$, V is unique [20]. Furthermore, an optimal μ^* must satisfy,

$$\begin{aligned} \text{support}(\mu^*(\cdot|s)) \subseteq \arg\min \Big[c(\lambda, s, u) - \\ \beta + \sum_{s'} p(s, u, s')V(s')\Big] \quad \forall s \in \mathcal{S}. \end{aligned} \tag{12}$$

By using standard arguments (see, e.g., [22]), it follows that the constrained problem has a stationary, though possibly randomized, optimal policy which is also optimal for the unconstrained problem considered in (10) for a particular choice of $\lambda = \lambda^*$ (say). In fact, we know from [22] that the optimal stationary policy can be taken to be *deterministic* for all but at most one s, i.e., there exists a unique $u^*(s)$ such that $\mu^*(u^*(s)|s) = 1$ and u^* is the solution to the following equation,

$$\begin{aligned} u^*(s) = \arg\min \Big[c(\lambda^*, s, u) - \\ \beta + \sum_{s'} p(s, u, s')V(s')\Big] \quad \forall s \in \mathcal{S}. \end{aligned} \tag{13}$$

Furthermore, for the single (if any) state s for which this fails, $\mu(\cdot|s)$ is supported on exactly two points. The optimal average cost β gives the minimum power consumed $\bar{P}^*$ subject to the specified delay (or equivalent queue length) constraint. Moreover, the following *saddle point condition* holds:

$$L(\mu^*, \lambda) \leq L(\mu^*, \lambda^*) \leq L(\mu, \lambda^*). \tag{14}$$

For a fixed λ, the Relative Value Iteration Algorithm (RVIA) is a known algorithm for solving the dynamic programming equation for the unconstrained problem in an iterative fashion. The average cost RVIA for determining the value function such that (11) is satisfied can be written as:

$$\begin{aligned} V_{n+1}(s) = \min_{u \in U(s)} [c(\lambda, s, u) + \\ \sum_{s'} p(s, u, s')V_n(s')] - V_n(s^0), \end{aligned} \tag{15}$$

where $s, s', s^0 \in \mathcal{S}$ and s^0 is any fixed state. $V_n(\cdot)$ is an estimate of the value function after n iterations for a fixed LM λ.

RVIA (15) requires the knowledge of transition probabilities $p(s, u, s')$ which in turn requires the knowledge of channel state and packet arrival distributions which are unknown. In the rest of the paper, we address this limitation by proposing a new approach based on post-decision state. We begin by first introducing the concept of a post-decision state.

IV. Post-Decision State Framework

We define the post-decision state[3] to be the virtual state of the system immediately *after* taking a decision but *before* the action of the noise. Let $s = (q, x)$ be the state S of the system in some time slot and the transmitter transmits $U = u$ packets, then the post-decision state denoted by $\tilde{S},\ \tilde{S} \in \mathcal{S}$ is $(q-u, x)$. If $A = a$ arrivals occur in the post-decision state and the channel gain changes to $X = x'$, then the system reaches the next actual system state, which can also be called the *pre-decision* state, $(q', x') = (q - u + a, x')$.

Let $\tilde{V} : S \to \mathcal{R}$ be the value function based on the post-decision state given by

$$\tilde{V}(\tilde{s}) = \mathbf{E}^s\big[V(S)\big].$$

where $\mathbf{E}^s$ is the expectation taken over all the pre-decision states that can be reached from the post-decision state $\tilde{s}$. Let ζ be the law for the arrivals and $\kappa(\cdot|\cdot)$ the transition probability function for the channel state process. Then $\tilde{V}$ satisfies the post-decision dynamic programming equation: for $\tilde{s} = (q, x)$,

$$\tilde{V}(\tilde{s}) = \sum_{a,x'} \zeta(a)\kappa(x'|x)\big(\min_{u \le q+a}[c(\lambda, (q+a, x'), u) + \tilde{V}((q+a-u, x'))]\big) - \beta. \tag{16}$$

From (15) and (16), we get the 'one component at a time' RVIA based on post-decision state as follows. Fix $\tilde{s}^0$. If the post-decision state at time n is $\tilde{s} = (q, x)$, then do:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \sum_{a,x'} \zeta(a)\kappa(x'|x)\big(\min_{u \le q+a}[c(\lambda, (q+a, x'), u) \\ &\quad + \tilde{V}_n((q+a-u, x'))]\big) - \tilde{V}_n(\tilde{s}^0); \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s}'')\ \ \forall\ \tilde{s}'' \ne \tilde{s}. \end{aligned} \tag{17}$$

The important thing to note here is that we update only the $\tilde{s}$-th component, not the rest. This is to lay ground for the on-line scheme we propose below, which is perforce 'one at a time', because one learns only about the current state being observed, and can, therefore, update only the corresponding component.

V. On-Line Algorithm

In this section, we propose an on-line algorithm to evaluate $\tilde{V}$. We note that the RVIA (15) is not amenable to on-line implementation because of the occurrence of *min* operator outside the averaging operation w.r.t. an unknown conditional law. On the other hand, (17) has a useful structure in the sense that the *expectation* operation has been moved outside of the *min* operator. The *expectation* can thus be dropped by performing averaging in time in order to determine the optimal value function.

[3] Similar ideas have been around in the literature before, see, e.g., [23], [24].

Let $f(n)$ be a positive update sequence that has the following properties,

$$\sum_n f(n) = \infty; \quad \sum_n (f(n))^2 < \infty. \tag{18}$$

Then, following the theory of stochastic approximation [25], we can remove the expectation from (17), and perform averaging via the following update equation. Recall that $S_n = (Q_n, X_n), n \ge 0$, is our state process. If $\tilde{s} = (q, x)$, the post-decision state at time n, then do:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= (1 - f(n))\tilde{V}_n(\tilde{s}) + f(n)\Big\{ \\ &\quad \min_u[c(\lambda, (q + A_{n+1}, X_{n+1}), u) \\ &\quad + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}^0)\Big\}, \\ &= \tilde{V}_n(\tilde{s}) + f(n)\Big\{ \min_u[c(\lambda, (q + A_{n+1}, X_{n+1}), u) \\ &\quad + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}) \\ &\quad - \tilde{V}_n(\tilde{s}^0)\Big\}, \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s})\ \ \forall\ \tilde{s}'' \ne \tilde{s}. \end{aligned} \tag{19}$$

The algorithm (19) is a primal RVIA scheme that attempts to solve the dynamic programming equation for a fixed value of the LM λ.

Let $e(n)$ be a positive update sequence that has the same properties as $f(n)$ expressed in (18). To reach the saddle point of the Lagrangian in (14), we introduce the following LM iterations,

$$\lambda_{n+1} = \Lambda[\lambda_n + e(n)\,(Q_n - \delta)], \tag{20}$$

where we use the projection operator Λ to project the LM onto interval $[0, L]$ for large enough $L > 0$, to ensure boundedness of the LM. We impose the following additional requirements on the update sequences $f(n)$ and $e(n)$,

$$\sum_n (f(n)^2 + e(n)^2) < \infty, \quad \lim_{n\to\infty} \frac{e(n)}{f(n)} \to 0. \tag{21}$$

The complete primal-dual RVI algorithm can be expressed as: for $S_n = \tilde{s} = (q, x)$,

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \tilde{V}_n(\tilde{s}) + f(n) \\ &\quad \Big\{ \min_u[c(\lambda_n, (q + A_{n+1}, X_{n+1}), u) \\ &\quad + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}) \\ &\quad - \tilde{V}_n(\tilde{s}^0)\Big\}, \end{aligned} \tag{22}$$

$$\tilde{V}_{n+1}(\tilde{s}'') = \tilde{V}_n(\tilde{s})\ \ \forall\ \tilde{s}'' \ne \tilde{s}, \tag{23}$$

$$\lambda_{n+1} = \Lambda[\lambda_n + e(n)\,(Q_n - \delta)]. \tag{24}$$

That these iterates, indeed, converge to the optimal values is proved in the Appendix. In (22, 23, 24), iterating simultaneously on the primal variable as well as the dual variable on different *timescales* ensures that the update rates of the primal and dual variables are different. The dual variable is updated on a slower timescale than the primal variable. This means that as viewed from the slower LM timescale, the primal variable appears to be equilibriated or converged to the optimal value corresponding to the current value of LM, while as viewed from the faster value function timescale, the LM values appear to be almost constant. This can be interpreted as iterating the LM after every $k_n = \frac{f(n)}{e(n)} >> 1$ iterations of the value function. Note that separation of timescales introduces a 'leader-follower' behaviour among the two components (fast and slow) of the algorithm which prevents the possible interference of one in the convergence of the other if they were run concurrently on the same timescale. We prove in the Appendix that this scheme converges 'almost surely' (a.s.).

Based on on-line primal-dual RVI computations (22, 23, 24), the transmitter implements the scheduling scheme as explained in Algorithm 1. We assume that the transmitter is aware of the value of channel state X_n in each time slot n. In practice, this may be achieved by receiver first estimating the channel state and then informing this to the transmitter through a feedback mechanism. In each time slot, the transmitter determines the number of packet arrivals, channel state and current queue length. The number of packets to be transmitted is then determined as explained in Algorithm 1. The value functions and the LM are then appropriately updated. The algorithm thus continues in each slot n.

Initialize the value function matrix $\tilde{V}(\tilde{s}) = \tilde{V}(q,x) \leftarrow 0 \ \forall q \in \mathcal{Q}, x \in \mathcal{X}$, the LM $\lambda \leftarrow 0$, the slot counter $n \leftarrow 1$, queue length $q \leftarrow 0$, channel states $x \leftarrow 0, x' \leftarrow 0$. Let reference state $\tilde{s}^0 = (0, x^1)$, where $x^1 \in \mathcal{X}$.
while TRUE **do**
  Determine number of arrivals $A_{n+1} = a$ and channel state $X_{n+1} = x'$ in the current slot.
  Transmit u packets, such that u minimizes the r.h.s in (22), thereby, power $P(x', u)$ required to transmit u packets is also determined (using (3)).
  Update the component $\tilde{s} = (q, x)$ of the value function matrix using (22). Rest of the components of the matrix remain unchanged.
  Update the LM λ using (24) ($Q_n = q$).
  Set $n \leftarrow n+1$, $q \leftarrow q + a - u, x \leftarrow x'$.
end while

Algorithm 1: The On-Line Algorithm

Remark 1: While the theoretical convergence is proved in the Appendix, our simulation results demonstrate that convergence of the algorithm occurs in reasonable number of iterations (time slots) for practical purposes. In long file transfer applications, the duration of transfer is of the order of seconds, while the slot duration in wireless systems is of the order of milli-seconds. Hence, non-optimality may be there only for certain part of data transfer.

Remark 2: In practical scenarios, we may not wait for the actual convergence to take place, but would like to be within a prescribed neighborhood of the optimal solution with high probability. Results of [26] give a bound on the number of iterations required for the iterate to be within a given distance from the convergence point thereafter with a prescribed high probability.

VI. Simulation Results

We implement Algorithm 1 in a simulation environment using MATLAB where we simulate the single user scenario depicted in Figure 1. We simulate a time slotted system with slot duration of 1 msec. Although the algorithm does not depend on any distribution for the channel gain H, for the purposes of modeling, we simulate an i.i.d. Rayleigh channel across slots. For a Rayleigh model, channel state X is an exponentially distributed random variable with probability density function given by $f_X(x) = \frac{1}{\alpha} e^{-\frac{x}{\alpha}}$, where α is the mean of X. We discretize the channel into eight equal probability bins, with the boundaries specified by { (-∞, -8.47 dB), [-8.47 dB, -5.41 dB), [-5.41 dB, -3.28 dB), [-3.28 dB, -1.59 dB), [-1.59 dB, -0.08 dB), [-0.08 dB, 1.42 dB), [1.42 dB, 3.18 dB), [3.18 dB, ∞) }. We choose the channel state space to be $\mathcal{X}$ = { $x^1 = -13$ dB, $x^2 = -8.47$ dB, $x^3 = -5.41$ dB, $x^4 = -3.28$ dB, $x^5 = -1.59$ dB, $x^6 = -0.08$ dB, $x^7 = 1.42$ dB, $x^8 = 3.18$ dB}. This discretization of the state space of X has been justified in [10]. We know from (3) that the power required for transmitting u packets when the channel state is x is given by, $P(x, u) = \frac{N_0 W}{x}\left(2^{u\ell/W} - 1\right)$, where N_0 is the power spectral density of the additive white Gaussian noise and W is the received signal bandwidth. We assume $W = 5$ MHz and the product $W N_0$ to be normalized to 1. We simulate i.i.d. arrivals with a Poisson distribution with mean θ. This implies that the probability of generating j packets is given by $p(j) = e^{-\theta}\frac{\theta^j}{j!}$. We assume that packets are of equal size $\ell = 5000$ bits. We also assume that the transmitter can transmit 1 to 8 packets in a slot. In practice, it can correspond to transmission using the following modulation schemes - Binary Phase Shift Keying (BPSK), Quadrature Phase shift Keying (QPSK), 8-Quadrature Amplitude Modulation (QAM), 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM respectively. This is because these modulation schemes have a spectral efficiency of $1 - 8$ bits/sec/Hz, and with $W = 5$ MHz, slot duration of 1 msec and packet size $\ell = 5000$ bits, the transmitter can potentially transmit $1 - 8$ packets/slot respectively.

We perform several experiments to validate the analytical results. While the convergence results are asymptotic, we demonstrate that for all practical purposes, the quantities like LM, power and delay converge within reasonable number of iterations. Consequently, for long file transfers lasting several seconds, the algorithm essentially operates *optimally* for a vast majority of the file transfer duration. In all the experiments, we simulate the algorithm for $100,000$ time slots. For the LM and value function update, we choose $e(n) = \frac{10}{n}$ and $f(n) = \frac{1}{n^{0.7}}$.

Experiment 1: Convergence of LM for various delay constraints, arrival rates and channel gains: This experiment demonstrates the convergence behaviour of the LM λ. In each slot, arrivals are generated with Poisson distribution with mean $\theta = 2$ packets/msec, i.e., arrival rate is 10 Mbits/sec. We choose $\alpha = 0.4698$ (-3.28 dB). In each slot, we generate X using the exponential distribution with mean α. We determine the channel state based on the partition that contains $X = x$ as explained above. We then use Algorithm 1 to determine the

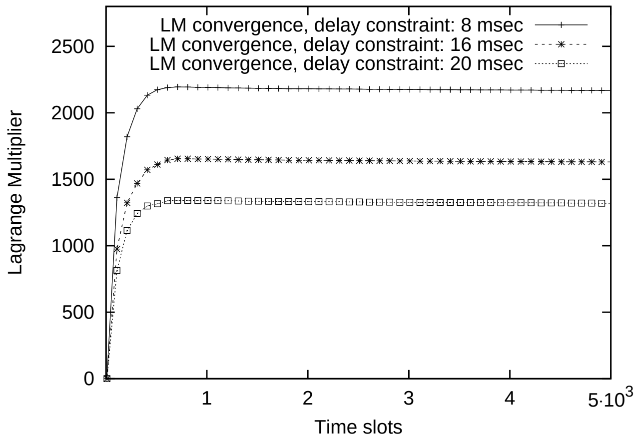

Fig. 2. Convergence of Lagrange multiplier for various average delay constraints

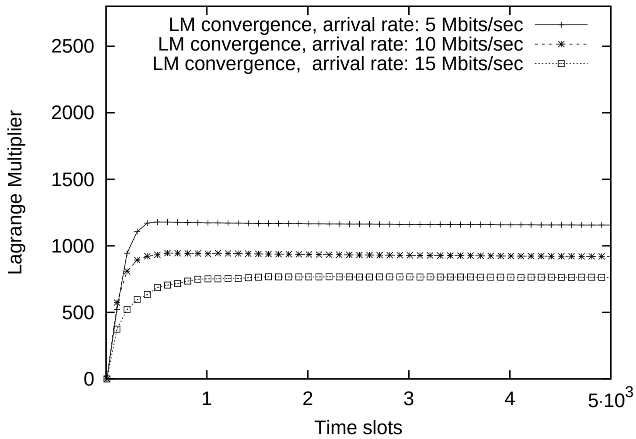

Fig. 3. Convergence of Lagrange multiplier for various average arrival rates

number of packets u that must be transmitted, the transmission power $P(x,u)$ and update the LM λ and the value function matrix $\tilde{V}$. We plot the variations in λ for delay constraints of 8 msec, 16 msec and 20 msec in Figure 2. From Figure 2, it can be observed that the LM converges in about 2000 slots. Each time slot may, typically, correspond to 1 msec. Thus convergence can be achieved in about 2 seconds.

We repeat this experiment for average arrival rates of 5 Mbits/sec, 10 Mbits/sec and 15 Mbits/sec with the delay constraint at 24 msec and the average channel state α to be 0.4698 (-3.28 dB). The convergence behaviour of LM λ for various arrival rates has been plotted in Figure 3. We further repeat the experiment with average channel state α as 0.4698, 0.6934 and 0.9817, i.e, -3.28 dB, -1.59 dB, and -0.08 dB. We keep the delay constraint at 24 msec and average arrival rate θ at 2 packets/msec, i.e., 10 Mbits/sec. The convergence behaviour of LM λ for various channel gains has been plotted in Figure 4. Figures 3 and 4 illustrate that the LM converges in approximately 2000 slots for all the arrival rate and channel gain variations.

Experiment 2: Convergence of delay and power for various delay constraints: In this experiment, we demonstrate the convergence behaviour of the average delay and the average power for various delay constraints. We determine the running

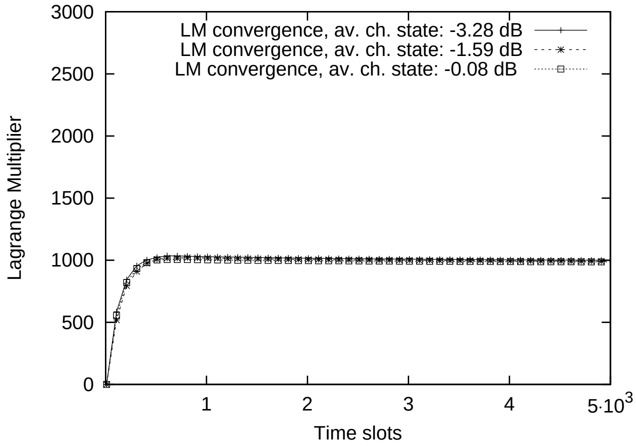

Fig. 4. Convergence of Lagrange multiplier for various average channel states

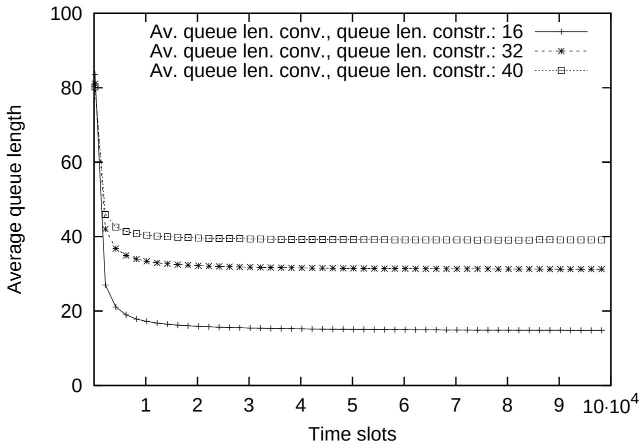

Fig. 5. Convergence of average delay for various average delay constraints

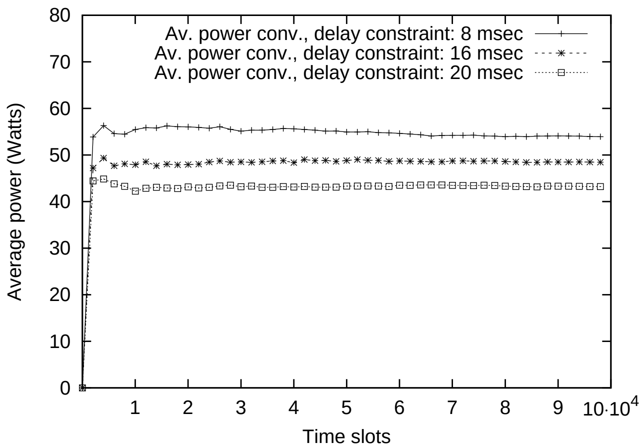

Fig. 6. Convergence of average power for various delay constraints

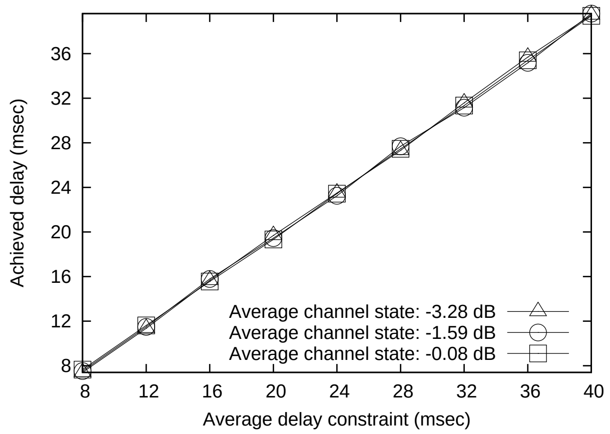

Fig. 7. Achieved system delay for various average delay constraints

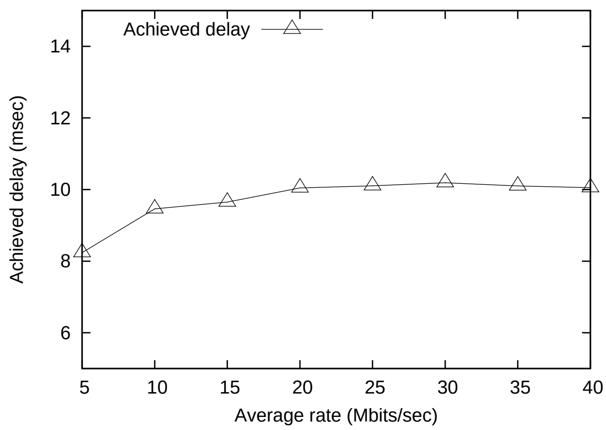

Fig. 9. Achieved system delay for various arrival rates

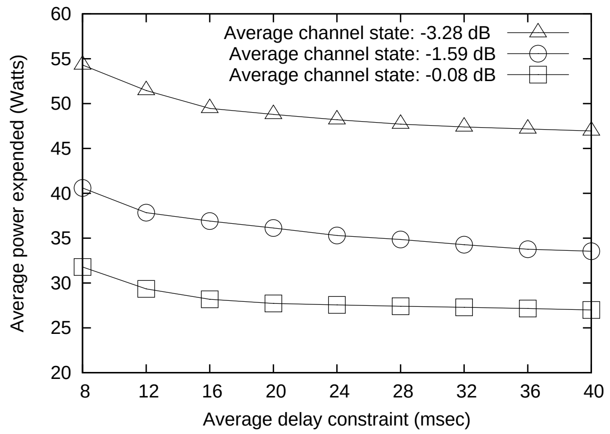

Fig. 8. Power-delay curve for various average channel states

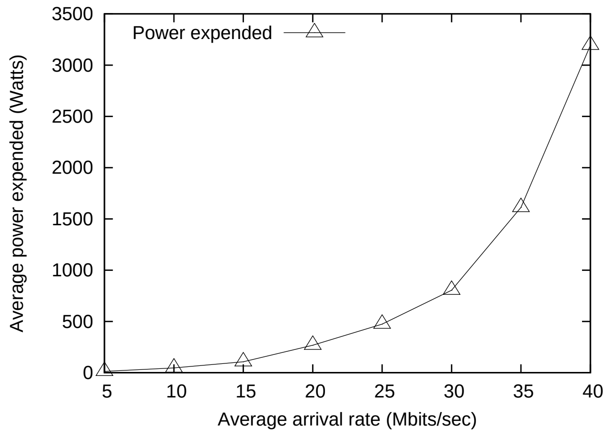

Fig. 10. Power-arrival rate curve

averages of the queue length Q_n^{av} and the power expended P_n^{av} in slot n as follows

$$Q_n^{av} = \frac{n-1}{n} Q_{n-1}^{av} + \frac{1}{n} Q_n,$$
$$P_n^{av} = \frac{n-1}{n} P_{n-1}^{av} + \frac{1}{n} P_n, \qquad (25)$$

with $Q_0^{av} = 0,\ P_0^{av} = 0$. The variations in Q_n^{av} and P_n^{av} with simulation time for the delay constraints of 8 msec, 16 msec and 20 msec are plotted in Figures 5 and 6 respectively. We keep the arrival rate θ at 2 packets/msec, i.e., 10 Mbits/sec and the average channel state α at 0.4698, i.e., -3.28 dB. Figures 5 and 6, illustrate that the average delay and average power converge reasonably fast.

Experiment 3: This experiment demonstrates that the algorithm satisfies various average delay constraints. We perform the simulations for the delay constraints 8, 12, 16, 20, 24, 28, 32, 36, 40 msec. We fix θ at 2 packets/msec, i.e., 10 Mbits/sec. We repeat the experiment for values of α = $0.4698, 0.6934, 0.9817$, i.e, -3.28 dB, -1.59 dB, -0.08 dB. From Figure 7, it can be observed that in all the cases, the average delay constraints are met. As the constraint on the delay increases, the average power required for transmission decreases as can be observed from Figure 8. The plot also demonstrates the convex characteristics of the power-delay curve that has been proved analytically in [8].

Experiment 4: This experiment demonstrates the range of arrival rates for which the algorithm satisfies a delay constraint of 10 msec. We perform the simulations for the arrival rates 5, 10, 15, 20, 25, 30, 35, 40 Mbits/sec by varying the mean θ of the Poisson distribution as 1, 2, 3, 4, 5, 6, 7, 8, packets/msec. We choose $\alpha = 0.4698$ (-3.28 dB). From Figure 9, it can be observed that the average delay constraint is met till the arrival rate becomes 8 packets/msec or 40 Mbits/sec. Beyond this, the arrival rate becomes more than the departure rate and thus the delay constraint cannot be satisfied as it violates the stability condition of the queue assumed in the formulation. As the arrival rate nears the capacity, the average power required for transmission is extremely high as can be observed from Figure 10.

VII. Conclusions and Discussions

In this paper, we have proposed an on-line learning algorithm for computing the optimal packet scheduling policy for single user fading channel, where the model is not known. The earlier works [8], [10], [11], [13], [14], [15] have provided valuable insights into the problem by proving the structural results under various assumptions (including that of more general state spaces). But none of them deals explicitly with the computation of optimal policy. We have proposed a new approach for on-line implementation of the optimal packet scheduling algorithm. This approach is based on reformulating the value iteration equation by introducing a virtual state called *post-decision state*. The resultant value iteration equation is amenable to on-line implementation by using stochastic approximation. Like other related work, we assume that the transmitter has a perfect knowledge of channel state, however, an explicit knowledge of the transition probability function of the channel state as well as the distribution of arrivals is not required for computing the policy. We have also proved that the on-line algorithm indeed converges to the optimal policy. Our simulation results have illustrated the performance of the algorithm under various scenarios and demonstrate that the algorithm can converge in reasonable number of slots for it to be practically useful with satisfaction of delay constraint.

In this paper, our focus has been on implementing on-line algorithm for single user fading channel. The problem of packet scheduling for multi-user fading channel under various assumptions has been explored in [27], [28], [29], [30]. However, extension of the problem considered in this paper (i.e., energy efficient delay constrained scheduling) for multi-user setting has not received much attention, notable exceptions being [21], [31]. We are currently investigating this problem and the applicability of the framework developed in this paper for its on-line implementation. Another issue that is worth investigating is how the structural properties of the optimal policy can be exploited to speed up the convergence time of the algorithm.

APPENDIX

Here we prove:

Theorem 1: For the algorithm (22, 23, 24), the iterates $(\tilde{V}_n, \lambda_n) \to (\tilde{V}, \lambda^*)$.

We prove this in several steps. First, note that the purpose of subtracting $\tilde{V}_n(\tilde{s}^0)$ from the r.h.s. in (22) is to keep the iterates stable. It turns out that $\tilde{V}_n(\tilde{s}^0)$ converges to the optimal average cost per stage β. More generally, we can replace $\tilde{V}_n(\tilde{s}^0)$ with a generic offset term $g(\tilde{V}_n)$ if we make the following assumption on the function $g : \mathbb{R}^{|S|} \to \mathbb{R}$ [32].

Assumption 1: $g(\cdot)$ is Lipschitz and for $\boldsymbol{\eta}$ equal to the constant vector of all 1's in $\mathbb{R}^{|S|}$, $g(\boldsymbol{\eta}) = 1$ and $g(\mathbf{x} + c\boldsymbol{\eta}) = g(\mathbf{x}) + c$ for $c \in \mathbb{R}$. We further assume that $g(a\mathbf{x}) = ag(\mathbf{x})$ for $a > 0$.

With Assumption 1, a generalized form of the primal-dual algorithm (22, 23, 24) can be written as follows. If the post-decision state at time n is $S_n = \tilde{s} = (q, x)$, then do:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \tilde{V}_n(\tilde{s}) + f(n) \\ &\quad \Big\{ \min_u [c(\lambda_n, (q + A_{n+1}, X_{n+1}), u) \\ &\quad + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}) \\ &\quad - g(\tilde{V}_n(\tilde{s})) \Big\}, \qquad (26) \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s}) \;\; \forall \;\; \tilde{s}'' \neq \tilde{s}, \qquad (27) \\ \lambda_{n+1} &= \Lambda[\lambda_n + e(n)\,(Q_n - \delta)]. \qquad (28) \end{aligned}$$

We now proceed to show that the algorithm (26, 27, 28) tracks an associated Ordinary Differential Equation (ODE) as described later. Recall that ζ is the law for the arrivals and $\kappa(\cdot|\cdot)$ the transition probability function for the channel state process. Note that the algorithm is unaware of these, we consider them only for the purpose of analysis. Let $T_\lambda : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ be the map defined by,

$$\begin{aligned} (T_\lambda \tilde{V})(\tilde{s}) &= \sum_{a,x'} \kappa(x'|x)\zeta(a) \Big\{ \min_{u \le q+a} \big[c(\lambda, (q + a, x'), u) \\ &\quad + \tilde{V}_n(q + a - u, x') \big] \Big\}, \\ \tilde{s} &= (q, x) \in \mathcal{S}. \qquad (29) \end{aligned}$$

Define $T'_\lambda : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ by $T'_\lambda(\tilde{V}) = T_\lambda(\tilde{V}) - g(\tilde{V})\boldsymbol{\eta}$. The RVIA can be written as:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \tilde{V}_n(\tilde{s}) + f(n)[T_{\lambda_n}(\tilde{V}_n(\tilde{s})) - g(\tilde{V}_n(\tilde{s})) - \tilde{V}_n(\tilde{s}) \\ &\quad + M_{n+1}(\tilde{s})], \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s}) \;\; \forall \;\; \tilde{s}'' \neq \tilde{s}, \\ \lambda_{n+1} &= \Lambda[\lambda_n + e(n)\,(Q_n - \delta)], \qquad (30) \end{aligned}$$

where, for $\tilde{s} = (q, x)$,

$$\begin{aligned} M_{n+1}(\tilde{s}) = \min_{u \le q + A_{n+1}} & [c(\lambda_n, (q + A_{n+1}, X_{n+1}), u) \\ & + \tilde{V}_n(q + A_{n+1} - u, X_{n+1})] - T_{\lambda_n}(\tilde{V}_n(\tilde{s})). \end{aligned}$$

Let $\mathcal{F}_n$ denote the σ-algebra, $\sigma(S_m, A_m, U_m, m \le n), n \ge 0$. It can be verified that $\mathbf{E}[M_{n+1}|\mathcal{F}_n] = 0$. Consider

$$\dot{\tilde{V}}(t) = T'_\lambda(\tilde{V}(t)) - \tilde{V}(t). \qquad (31)$$

It can be argued as in [32] that as $t \to \infty$, $\tilde{V}(t)$ converges to the unique fixed point of $T'_\lambda(\cdot)$, i.e., $\tilde{V}^\lambda$ such that

$$T'_\lambda(\tilde{V}) = \tilde{V}^\lambda \qquad (32)$$

Hence $\tilde{V}^\lambda$ is the globally asymptotically stable equilibrium for the above ODE.

Lemma 1: The post-decision value function iterates $\{\tilde{V}_n\}$ remain bounded a.s.

Proof: Consider $T^0 : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ defined by,

$$(T^0\tilde{V})(\tilde{s}) = \sum_{a,x'} \kappa(x'|x)\zeta(a) \min_{u \le q+a} [\tilde{V}(q+a-u, x')] - g(x)\boldsymbol{\eta}. \qquad (33)$$

Then T^0 is also a contraction w.r.t $||\cdot||_w$, $\lim_{c\to\infty} \frac{T'_\lambda(c\tilde{V})}{c} = T^0(\tilde{V})$, and the ODE,

$$\dot{\tilde{V}}(t) = T^0(\tilde{V}) - \tilde{V}, \qquad (34)$$

has origin as the globally asymptotically stable equilibrium (again, by arguments of [33]). This is the scaled limit of the ODE (34) in the sense of [34]. Note that it is independent of λ. The claim follows from Theorem 2.1 of [34]. ■

Lemma 2: In algorithm (26, 27, 28), $\tilde{V}_n - \tilde{V}^{\lambda_n} \to 0$ a.s., where $\tilde{V}^{\lambda_n}$ is the value function based on post-decision state for $\lambda = \lambda_n$.

Proof: The algorithm (30) is the standard stochastic approximation algorithm with martingale difference noise M_{n+1}. From (21) and (30) it can be seen that the LM is varied on a much slower timescale than the post-decision relative value function estimate $\tilde{V}_n$. Therefore, the post decision value function iterations *see* the LM to be almost constant. To be precise, the λ iterations in (30) can be written as, $\lambda_{n+1} = \lambda_n + \nu(n)$, where $\nu(n) = \mathcal{O}(e(n)) = o(f(n))^4$. Hence the limiting ODEs associated with (30) for analyzing the $\tilde{V}_n$ iterates are, $\dot{\tilde{V}}(t) = T'_\lambda(\tilde{V}(t)) - \tilde{V}(t); \quad \dot{\lambda}(t) = 0$. Since $\dot{\lambda}(t) = 0$, for analyzing the $\tilde{V}_n$ iterates, it suffices to consider the ODE,

$$\dot{\tilde{V}}(t) \quad = \quad T'_\lambda(\tilde{V}(t)) - \tilde{V}(t), \tag{35}$$

for any prescribed value of the LM λ. The rest follows by standard arguments as in [35]. ■

The $\{\lambda_n\}$ iterations are bounded since they are constrained to remain in the interval $[0, L]$. We now prove that the coupled iterates converge to their optimal values $(\tilde{V}^{\lambda^*}, \lambda^*)$. Let $G(\lambda) \stackrel{\Delta}{=} \min_\mu \mathcal{L}(\mu, \lambda)$. We reproduce the following results from [36].

Lemma 3: G is piecewise linear and concave. In particular, it is continuously differentiable except at finitely many points where both right and left derivatives exist.

Define $h(\lambda) \stackrel{\Delta}{=} \sum_{s,u} \rho^\mu(s) c(\lambda, s, u)$, where μ is the optimal stationary policy when LM λ is used (note that this introduces an additional $\lambda-$dependence not explicitly shown). Consider

$$\dot{\lambda}(t) = h(\lambda(t)) - \overline{q}, \tag{36}$$

constrained to remain in $(0, \infty)$.

Lemma 4: Equation (36) is same as the gradient ascent,

$$\dot{\lambda}(t) = \nabla G(\lambda(t)), \tag{37}$$

interpreted in the Caratheodory sense, i.e., as the integral equation,

$$\lambda(t) = \lambda(0) + \int_0^t \nabla G(\lambda(s)) ds, \quad t \geq 0. \tag{38}$$

Proof: This follows using the ‘generalized envelope theorem’ as in [36]. ■

Corollary 1: The iterates λ_n a.s. converge to the set of maxima of G.

This follows by standard arguments as in [36].

Corollary 2: $\{\mu_n\}$ converge to the set of optimal policies corresponding to $\lambda \in \arg\max(G)$, a.s.

Note that any $\lambda \in \arg\max(G)$ is a valid Lagrange multiplier. This completes the proof of the theorem.

REFERENCES

[1] T. S. Rappaport, A. Annamalai, R. Buehrer, and W. H. Tranter, “Wireless Communications: Past Events and a Future Perspective,” *IEEE Communications Magazine*, vol. 40, no. 5, pp. 148–161, 2002.

[2] D. P. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1987.

[3] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-Layer Design for Wireless Networks,” *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, 2003.

[4] R. Berry and E. M. Yeh, “Cross-Layer Wireless Resource Allocation,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 59–68, 2004.

[5] E. Biglieri, J. Proakis, and S. Shamai, “Fading Channels: Information-Theoretic and Communications Aspects,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2619–2692, 1998.

[6] A. Goldsmith and P. Varaiya, “Capacity of Fading Channels with Channel Side Information,” *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1986–1992, 1997.

[4] $\mathcal{O}$ and o stand for the “Big-oh” and “Small-oh” notation respectively. Intuitively, this means that $\nu(n)$ goes faster to 0 than $f(n)$.

[7] D. Tse and S. Hanly, “Multi-access Fadaing Channels - Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities,” *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, 1998.

[8] R. A. Berry and R. G. Gallager, “Communication over Fading Channels with Delay Constraints,” *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.

[9] R. Berry, “Power and Delay Trade-offs in Fading Channels,” June 2000. PhD Thesis, Massachusetts Institute of Technology.

[10] H. Wang and N. Mandayam, “A Simple Packet Transmission Scheme for Wireless Data over Fading Channels,” *IEEE Transcations on Communications*, vol. 52, no. 7, pp. 1055–1059, 2004.

[11] M. Goyal, A. Kumar, and V. Sharma, “Power Constrained and Delay Optimal Policies for Scheduling Transmissions over a Fading Channel,” in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 311–320, Mar. 2003.

[12] D. Rajan, A. Sabharwal, and B. Aazhang, “Delay-Bounded Packet Scheduling of Bursty Traffic over Wireless Channels,” *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 125–144, 2004.

[13] D. V. Djonin and V. Krishnamurthy, “Structural Results on the Optimal Transmission Scheduling Policies and Costs for Correlated Sources and Channels,” in *IEEE CDC*, pp. 3231–3236, Dec. 2005.

[14] M. Agarwal, A. Karandikar, and V. S. Borkar, “Structural Properties of Optimal Transmission Policies over a Randomly Varying channel,” *Accepted for publication, IEEE Transactions on Automatic Control*, 2007.

[15] G. Rajadhyaksha and V. S. Borkar, “Transmission Rate Control over Randomly Varying Channels,” *Probability in the Engineering and Informational Sciences*, vol. 19, pp. 73–82, Jan. 2005.

[16] B. Prabhakar, E. U. Biyikoglu, and A. E. Gamal, “Energy-Efficient Transmission over a Wireless Link via Lazy Packet Scheduling,” in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 386–394, Apr. 2001.

[17] B. E. Collins and R. L. Cruz, “Transmission Policies for Time Varying Channels with Average Delay Constraints,” in *Proceedings of 1999 Allerton Conference on Communication, Control, and Computation*, 1999.

[18] E. Altman, *Constrained Markov Decision Processes*. Boca Raton, FL: Chapman and Hall/CRC Press, 1999.

[19] D. Farias and B. V. Roy, “The Linear Programming Approach to Approximate Dynamic Programming,” *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.

[20] M. Puterman, *Markov Decision Processes*. New York: Wiley, 1994.

[21] M. J. Neely, “Optimal Energy and Delay Tradeoffs for Multi-user Wireless Downlinks,” in *Proceedings of IEEE INFOCOM*, pp. 1–13, 2006.

[22] V. S. Borkar, “Convex Analytic Methods in Markov Decision Processes,” in *E. A. Feinberg, A. Schwartz (Eds.) Handbook of Markov Decision Processes*, (Dordrechet), pp. 347–375, Kluwer Academic Publishers, 2001.

[23] V. F. Magirou, “Stockpiling under Price Uncertainty and Storage Capacity Constraints,” *European Journal of Operational Research*, pp. 233–246, 1982.

[24] B. V. Roy, D. P. Bertsekas, Y. Lee, and J. N. Tsitsiklis, “A Neuro-Dynamic Programming Approach to Retailer Inventory Management,” in *Proceedings of IEEE CDC*, vol. 4, pp. 4052–4057, 1997.

[25] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[26] V. S. Borkar, “On the Lock-in Probability of Stochastic Approximation,” *Combinatorics, Probability and Computing*, vol. 11, no. 1, pp. 11–20, 2002.

[27] X. Liu, E. Chong, and N. Shroff, “Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks,” *IEEE Journal on Selected Areas in Communication*, vol. 19, no. 10, pp. 2053–2064, 2001.

[28] E. F. Chaponniere, P. Black, J. M. Holtzman, and D. Tse, “Transmitter directed Multiple Receiver System using Path Diversity to Equitably Maximize Throughput ,” Sept. 2002. U.S. Patent No. 6449490.

[29] S. Shakkottai and A. Stolyar, “Scheduling for Multiple Fows Sharing a Time-varying Channel: the Exponential Rule,” *AMS Translations Series 2*, vol. 207, 2002.

[30] A. Bhorkar, A. Karandikar, and V. S. Borkar, “Power Optimal Opportunistic Scheduling,” in *Proceedings of IEEE GLOBECOM*, pp. 1–5, Nov. 2006.

[31] E. Yeh and A. Cohen, “Delay Optimal Rate Allocation in Multiaccess Fading Communications,” in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pp. 404–407, Oct. 2002.

[32] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning Algorithms for Markov Decision Processes with Average Cost," *SIAM Journal on Control and Optimization*, vol. 40, pp. 681–698, 2001.
[33] V. S. Borkar and K. Soumyanath, "A New Analog Parallel Scheme for Fixed Point Computation, part 1: Theory," *IEEE Transactions on Circuits and Systems I*, vol. 44, pp. 351–355, 1997.
[34] V. S. Borkar and S. P. Meyn, "The ODE Method for Convergence of Stochastic Approximation and Reinforcement Learning," *SIAM Journal on Control and Optimization*, vol. 38, pp. 447–469, 2000.
[35] V. S. Borkar, "Stochastic Approximation with Two Time Scales," *Systems and Control Letters*, vol. 29, pp. 291–294, 1996.
[36] V. S. Borkar, "An Actor-Critic Algorithm for Constrained Markov Decision Processes," *Systems and Control Letters*, vol. 54, pp. 207–213, 2005.

Vivek S. Borkar Vivek S. Borkar obtained his B.Tech. in Electrical Engineering from Indian Institute of Technology, Bombay, in 1976, M.S. in Systems and Control from Case Western Reserve University in 1977 and Ph.D. in Electrical Engineering and Computer Science from University of California, Berkeley, in 1980. He has held positions at TIFR Centre, Bangalore, Indian Institute of Science, Bangalore, and is currently with the School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai. He has also held visiting positions at University of Twente, MIT, University of California, Berkeley, and University of Maryland, College Park. He has authored three books and over a hundred archival publications, and is the recipient of several national and international honours, including the Fellowships of IEEE, various scientific academies in India, Homi Bhabha and J. C. Bose Fellowships, S. S. Bhatnagar Award from the Govt. of India. His research interests are in theory, algorithms and applications of stochastic optimization.

Nitin Salodkar Nitin Salodkar obtained his Bachelor of Engineering in Computer Science and Engineering from Visvesvaraya National Institute of Technology, Nagpur, India in 2000. He worked with Motorola India Electronics Pvt. Ltd. India for about two years. Currently he is a Ph.D. candidate in the Department of Computer Science and Engineering at IIT Bombay. His research interests are in the area of Communication Networks, specifically, cross layer design in wireless networks.

Abhijeet Bhorkar Abhijeet Bhorkar obtained his B.Tech. and M.Tech. degrees in Electrical Engineering from Indian Institute of Technology, Bombay in 2006. He is currently working toward the Ph.D. degree at the University of California, San Diego. His research interests are primarily in the areas of stochastic control and estimation theory, information theory, and their applications in the optimization of wireless communication systems.

Abhay Karandikar Abhay Karandikar obtained his M.Tech. and Ph.D degrees in Electrical Engineering from Indian Institute of Technology, Kanpur in 1988 and 1994, respectively. During 1994-97, he worked at the Centre for Development of Advanced Computing (C-DAC), an Indian Government initiative in supercomputing. As Team Co-ordinator, he led the team on high-speed interconnection network design of the PARAM 9000 supercomputer, and several other high-speed networking projects in C-DAC. He joined the Department of Electrical Engineering, IIT Bombay in April 1997, where he is currently a Professor. He co-founded the venture backed company Eisodus Networks focussing on broadband access to cater to emerging India- and Asia-Pacific markets. His research interests include quality of service guarantees in communications networks, broadband wireless networks, and digital system design. He has consulted extensively for industries in the areas of communications and networking over the past decade. His research group at IIT Bombay is currently focused on cross-layer design issues in wireless networks.