Optimal Scheduling over Time-Varying Channels with Traffic Admission Control: Structural Results and Online Learning Algorithms

Khoa T. Phan, *Student Member, IEEE*, Tho Le-Ngoc, *Fellow, IEEE*, Mihaela van der Schaar, *Fellow, IEEE*, and Fangwen Fu, *Associate Member, IEEE*

***Abstract*—This work studies the joint scheduling- admission control (SAC) problem for a single user over a fading channel. Specifically, the SAC problem is formulated as a constrained Markov decision process (MDP) to maximize a utility defined as a function of the throughput and queue size. The optimal throughput- queue size trade-off is investigated. Optimal policies and their structural properties (i.e., monotonicity and convexity) are derived for two models: simultaneous and sequential scheduling and admission control actions. Furthermore, we propose online learning algorithms for the optimal policies for the two models when the statistical knowledge of the time-varying traffic arrival and channel processes is unknown. The analysis and algorithm development are relied on the reformulation of the Bellman's optimality equations using suitably defined state-value functions which can be learned online, at transmission time, using time-averaging. The learning algorithms require less complexity and converge faster than the conventional Q-learning algorithms. This work also builds a connection between the MDP based formulation and the Lyapunov optimization based formulation for the SAC problem. Illustrative results demonstrate the performance of the proposed algorithms in various settings.**

***Index Terms*—Scheduling, traffic admission control, Markov decision process (MDP), learning, structural results.**

I. Introduction

On the communications over time-varying channels, when the probability distribution functions (PDFs) of the channel and traffic arrival processes are known a-priori, optimal scheduling policies can be computed off-line [1]–[5], for instance by using dynamic programming techniques. However, such statistical knowledge is often unavailable a-priori in real-life communications, and hence, developing online scheduling algorithms without requiring known PDFs is important [6]–[9]. While the mentioned works have addressed these issues for the scheduling problem *without* traffic admission control, our current work studies the joint scheduling- traffic admission control (SAC) problem where only a portion of the arriving traffic can be buffered for transmission. In particular, we first analyze the structural properties of the optimal SAC policies and then propose online learning algorithms for the optimal policies under a-priori unknown PDFs.

In the scheduling without admission control, the central concept is the power- delay trade-off [1]. That says, a delay (or an average congestion) requirement can be attained by increasing the transmission power, i.e., increasing the service rate. However, when there is a constraint on the maximum transmission power, a delay bound might be impossible to achieve. One solution is to implement admission control to limit the traffic entering the buffer by admitting only a portion of the arrival traffic. Also, admission control is especially required to ensure queue stability (finite queue length) when the power budget is smaller than the minimum power required to stabilize the queue without admission control. It can be observed that there is a trade-off between maximizing the throughput and minimizing the average queue size (or average congestion). Hence, admission control can be viewed as shaping the arrivals from some external arriving sources to achieve some trade-off outcomes. The works [10], [11] propose the energy constrained control algorithm (ECCA) to stabilize the queue and maximize the throughput using Lyapunov optimization theory. Although simple, ECCA cannot achieve optimal throughput- queue size trade-off because it does not learn the system dynamics. Also, the derived bounds are only tight for sufficiently high traffic loading while the tightness is not known for light traffic loading. Alternatively, by exploiting a Markov decision process (MDP) approach and stochastic control tools, this work focuses on the control policies achieving the *optimal* trade-off in *all* traffic loading regions. The proposed algorithms learn the system dynamics and adapt the control decisions accordingly.

This work formulates the SAC problem as a constrained MDP to maximize a utility defined as the difference between the *throughput benefit* and the *buffer cost* (or congestion cost). The benefit and cost are increasing functions of the throughput and the buffer size, respectively. Such utility functions capture the inherent trade-off between maximizing

Manuscript received October 3, 2012; revised February 28, 2013; accepted June 3, 2013. The associate editor coordinating the review of this paper and approving it for publication was P. Wang.

K. T. Phan and T. Le-Ngoc are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada H3A 0E9 (e-mail: khoa.phan@mail.mcgill.ca; tho.le-ngoc@mcgill.ca).

M. van der Schaar is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: mihaela@ee.ucla.edu).

F. Fu is with Intel, Folsom, CA, 95630, USA (e-mail: fu.fangwen@gmail.com).

In this work, K. T. Phan acknowledges the partial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), McGill Engineering Doctoral Award, SR Telecom Doctoral Fellowship, and NSF CNS 0831549 Grant; T. Le-Ngoc acknowledges the partial support of the NSERC; and M. van der Schaar and F. Fu acknowledge the partial support by the NSF CNS 0831549 Grant.

Part of this paper was presented at the IEEE International Conference on Communications (ICC), Budapest, Hungary, June 2013.

Digital Object Identifier 10.1109/TW.2013.081913.121525

1536-1276/13$31.00 © 2013 IEEE

the throughput and minimizing the queue size: The larger the throughput, the larger the buffer cost becomes and vice versa. One possible application of the model is delay-sensitive loss-tolerant multimedia transmission where traffic with different priorities and delay deadlines [14], [15] is transmitted. In such systems, lower priority traffic can be dropped to ensure that the higher priority traffic meets the delay deadlines, especially for power-limited systems. We emphasize that due to the time-varying nature of the channel and traffic arrival processes, admission control needs to be done intelligently to balance the throughout and queue size, especially when the statistics of the random processes are a-priori unknown. To address this issue, this work develops learning algorithms for the optimal SAC policies using stochastic approximation without requiring explicit knowledge on the PDFs. While stochastic approximation based learning algorithms are proposed in [7], [8] for the scheduling problem *without* admission control, optimal learning for the SAC policies has never been addressed.

This work establishes the increasing concavity of the optimal throughput- queue size trade-off in SAC, which complements the decreasing convexity of the optimal power- delay trade-off in the scheduling without admission control [1]. The structural results of the optimal policies under two control models: simultaneous and sequential scheduling and admission control actions are derived. To help deriving the structural results, we define various state-value functions, which are used to rewrite the Bellman's optimality dynamic programming equations. Furthermore, to learn the optimal policies under unknown system dynamics, it is sufficient to learn these value functions, which is shown to require less storage complexity and converge faster than learning the Q-function as in the conventional Q-learning algorithms [6], [16]. We also observe that the ECCA in [10] can be interpreted as an approximate learning algorithm where it approximates the optimal concave value functions by linear functions. Hence, the proposed learning approach builds a connection between MDP formulation and Lyapunov optimization based formulation.

The remaining of the manuscript is organized as follows. Section II introduces the system model and formulates the optimal SAC problem. The optimal throughput- queue size trade-off is presented. Section III and IV analyze the structural results of the optimal policies and propose online learning algorithms for two control models. Numerical results are presented in Section V. All proofs are relegated to the end of the manuscript.

II. Optimal Joint Scheduling-Admission Control

A. System Description

We consider a SAC model where a single user (a transmitter-receiver pair) transmits data stored in a buffer over a fading channel. Time is divided into slots of equal duration. The dynamics of the buffer (or queue) is controlled using admission control and scheduling actions. Specifically, in each slot, the scheduling action computes the amount of traffic removed from the buffer for transmission to the receiver. Also, the admission control action determines the amount of traffic (from the newly-arriving traffic) to be stored into the buffer. Under a maximum constraint on the power consumption (and hence, transmission rate), it is clear that there are two conflicting objectives. One objective is to maximize the traffic throughput (or the average traffic admission rate to the buffer). The second objective is to minimize the average queue size (or the average congestion). We now describe the system components in details.

The wireless channel is assumed to be block-fading over the time slots. Denote h^t as the channel state representing the (normalized) power gain in slot t, $t = 0, 1, \ldots$. We assume:

(A1) The channel process $\{h^t\} \in \mathcal{H}$ is independent and identically distributed (i.i.d.) block-fading with general PDF $p_{\mathcal{H}}(h^t)$ over the finite channel state space $\mathcal{H}$.

Denote $\mathcal{B} \in [0, \infty)$[1] and $b^t \in \mathcal{B}$ as the queue state space and the queue state representing the queue size (in number of bits) in slot t, respectively. Let a^t, $a^t \in [0, b^t]$ (in number of bits) denote the scheduling action in slot t. Moreover, let y^t and r^t, $r^t \in [0, y^t]$ (in number of bits) represent the amount of new arrivals and the amount of arrivals admitted into the buffer in slot t. We assume:

(A2) The traffic arrival process $\{y^t\} \in \mathcal{Y} = [0, y_{\max}]$ is i.i.d. over slots with general PDF $p_{\mathcal{Y}}(y^t)$.

Given b^0 as the initial backlog, the queue dynamics across time slots satisfy the Lindley's recursion:

$$b^{t+1} = [b^t - a^t]^+ + r^t \tag{1}$$

where $[x]^+$ denotes $\max\{x, 0\}$. Note that without admission control, $r^t = y^t, \forall t$. Also, the arrival traffic in slot t can only be scheduled in the next slot earliest.

The reliable transmission of a (in number of bits) under channel state h incurs a power $c(h, a)$.[2] We assume:

(A3) The power functions $c(h, a)$ are strictly convex increasing differential with a; strictly decreasing with h; $c(h, 0) = 0$, and $\lim\limits_{a \to \infty} c(h, a) = \infty$.

We define the (sample path dependent) throughput R as $R \triangleq \liminf\limits_{t \to \infty} \dfrac{1}{t} \mathbb{E}\left\{\sum\limits_{\tau=0}^{t-1} r^\tau\right\}$ where the expectation operator $\mathbb{E}\{.\}$ is taken over the probability measure induced by the random processes and some SAC policy (to be defined later). The (average) queue size and power consumption are, respectively, $B \triangleq \limsup\limits_{t \to \infty} \dfrac{1}{t} \mathbb{E}\left\{\sum\limits_{\tau=0}^{t-1} b^\tau\right\}$ and $C \triangleq \limsup\limits_{t \to \infty} \dfrac{1}{t} \mathbb{E}\left\{\sum\limits_{\tau=0}^{t-1} c(h^\tau, a^\tau)\right\}$. It is assumed that the power C does not exceed a maximum value $C_{\max}$.

The utility obtained in slot t is defined as the difference between the *throughput benefit* obtained $f_b(r^t)$ and the *buffer cost* $f_c(b^t)$ incurred in the same slot, i.e., $u(r^t, b^t) \triangleq f_b(r^t) -$

[1] We allow the buffer to be an arbitrary real value for mathematical convenience [1], [8].

[2] One possible power function is derived from the Shannon theoretic function $c(h, a) = (2^a - 1)/h$ which will be used in the simulation section. This power function satisfies assumption (A3).

$f_c(b^t)$.[3] The (average) utility is defined as:

$$U \triangleq \liminf_{t\to\infty} \frac{1}{t}\mathbb{E}\left\{\sum_{\tau=0}^{t-1} f_b(r^\tau) - f_c(b^\tau)\right\}. \tag{2}$$

We make the following assumption:

(A4) The benefit function $f_b(r)$ is increasing concave differential with r; The cost function $f_c(b)$ is increasing convex differential with b.

The increasing monotonicity assumption of the functions is consistent with the fact that the throughput benefit and buffer cost increase, respectively, with the admission rate, and queue size. The assumed concavity of f_b and convexity of f_c describe the decreasing marginal utility improvement with throughput, and increasing marginal utility deterioration with delay, respectively as observed in some data applications, such as file transfer, voice transmission, and web browsing [17]. Such assumption has been widely used in literature, for example, see [3], [6], [8], [12], and references therein.

B. Optimal SAC Problem Formulation

The utility-optimal SAC problem can be posed as:

$$\max_{\pi\in\Pi} \quad U \quad \text{such that:} \quad C \leq C_{\max} \tag{3}$$

where Π is the set of all feasible (or admissible) SAC control policies π (to be defined in the following).

Observation. The SAC problem to maximize the throughput under the maximum queue size constraint in [10], [11] has similar Lagrangian function as (3) with $f_b(r) = r$ and $f_c(b) = \kappa b$ for some positive κ. Hence, it can also be studied using the formulation (3).

1) Optimal Throughput-Queue Size Trade-Off: To study the trade-off, in (3), we let the functions be $f_b(r) = r$ and $f_c(b) = \kappa b$ for some coefficients $\kappa \in [0, 1)$ which controls the trade-off.[4] The corresponding maximum objective value is $U^* = R^* - \kappa B^*$ where R^* and B^* are the throughput and (average) queue size. Since U^* is maximized, R^* is the maximum achievable throughput given that the queue size is equal to B^*.

More generally, now for any B, define $R(B)$ to be the maximum throughput such that the queue size is less than or equal to B. Hence, with this definition, we have $R(B^*) = R^*$. Proposition 1 characterizes the optimal trade-off $R(B)$.

Proposition 1: Under maximum power constraint, $R(B)$ is concave increasing of B.

PROOF: The proof based on sample path arguments is presented in Appendix A.

The points on the trade-off curve $R(B)$ are obtained by varying the coefficients $\kappa \in [0, 1)$.

TABLE I
TABLE OF NOTATIONS

Notations	Meanings
(b, h, y)	(Buffer state, channel state, arrival state)
(a, r)	(Scheduling action, admission control action)
$c(h, a)$	Power cost function
$f_b(r)$	Throughput benefit function
$f_c(b)$	Buffer cost function
$u = f_b(r) - f_c(b)$	Utility function
$J(b, h, y; \beta)$	Pre-decision value function
$J_{\text{dec}}(\check{b}; \beta)$	Post-decision value function
$\check{b}^t \triangleq [b^t - a^t]^+ + r^t$	Post-decision buffer state in slot t
$V(b, h; \beta)$	Pre-transmission value function
$V_{\text{tr}}(\hat{b}; \beta)$	Post-transmission value function
$V_{\text{ad}}(\tilde{b}; \beta)$	Post-admission value function
$\hat{b}^t \triangleq [b^t - a^t]^+$	Post-transmission buffer state in slot t
$\tilde{b}^t \triangleq \hat{b}^t + r^t$	Post-admission buffer state in slot t

Since the cost function $f_c(b)$ is unboundedly increasing with the queue size (assumption (A4)), the objective function in (3) is unboundedly decreasing with the queue size. Hence, the optimal solutions of (3) must result in a finite queue size, and hence, the underlying Markov chain is aperiodic and irreducible. Consequently, according to Theorem 12.7 in [18], the constrained MDP problem (3) admits an optimal solution that can be found using the Lagrangian approach:

$$\min_{\beta\geq 0} \left\{\max_{\pi\in\Pi} \left\{U - \beta C\right\} + \beta C_{\max}\right\}. \tag{4}$$

Therefore, to study (4) (and thus (3)), we can first study the inner maximization for a given positive multiplier β:

$$\max_{\pi\in\Pi} \left\{U - \beta C\right\}. \tag{5}$$

In the following sections, we study the optimal solutions of (5) for two SAC models: (i) simultaneous (in Section III) and (ii) sequential (in Section IV). While the former model is suitable when the newly arriving traffic is synchronized at the slot boundaries, the latter model is more applicable when the new traffic arrives asynchronously (or randomly) during a slot duration, e.g., after the scheduling decision instant at the beginning of each slot.[5]

III. Optimal Policies for Model 1: Structural Results and Online Learning Algorithm

In *Model 1*, it is assumed that in slot t, the controller observes the state (b^t, h^t, y^t) and determine the actions a^t and r^t simultaneously. This control model is similar to that considered in [4]. Hence, a stationary SAC policy π_1 can be represented by a 2-tuple function $(a, r) : \mathcal{B} \times \mathcal{H} \times \mathcal{Y} \to \mathbb{R}^+ \times \mathbb{R}^+$ specifying the control actions in slot t as $a^t = a(b^t, h^t, y^t) \in [0, b^t]$ and $r^t = r(b^t, h^t, y^t) \in [0, y^t]$ where $\mathbb{R}^+$ denotes the set of nonnegative numbers.

A. Post-Decision States and Post-Decision State-Value Function

Define $J(b, h, y; \beta)$ as the (pre-decision) state-value function, i.e., $J(b, h, y; \beta)$ is the optimal value of (5) with the

[3] The benefit and cost functions are in general application-dependent. For applications that are more delay sensitive, f_c is large, so that less traffic is admitted to keep the queue size small. For applications that are more sensitive to traffic losses, f_b is large, so that more traffic is admitted.

[4] Linear buffer cost model has been used in several works [4], [9], [12] and is related to the queuing delay by Little's theorem. Moreover, since it holds true that $R < B$, $\kappa \in (0, 1)$ to avoid triviality, otherwise, no traffic is admitted.

[5] For convenience, we have summarized the various notations used in this work in Table I.

starting state $(b^0, h^0, y^0) = (b, h, y)$. The Bellman's optimality dynamic programming equation for (5) is:

$$J(b,h,y;\beta) = \max_{\substack{a:a\leq b\\ r:r\leq y}} \Bigg\{ f_b(r) - f_c(b) - \beta c(h,a) + \sum_{h'\in\mathcal{H}} \sum_{y'\in\mathcal{Y}} p_{\mathcal{H}}(h')p_{\mathcal{Y}}(y')J(b-a+r,h',y';\beta)\Bigg\} - J(b_0,h_0,y_0;\beta). \quad (6)$$

for some arbitrary but fixed state (b_0, h_0, y_0). The optimal policy π_1^* is the optimal solution of (6). We can see that (6) requires known PDFs to evaluate the expectation. However, the PDFs are often unknown in real-time systems which makes the exact computation of the expectation impossible. Conventional reinforcement Q-learning algorithms [6], [16] can be used to learn π_1^* via learning the state-action Q function without requiring known PDFs.[6] However, Q-learning algorithms require large (storage) complexity, and exhibit slow convergence [8], [9]. We will propose an alternative approach with less complexity and faster convergence in the following.

Similar to [7]–[9], we define the post-decision state-value function $J_{\rm dec}(\check{b};\beta)$ as:

$$J_{\rm dec}(\check{b};\beta) = \sum_{h'\in\mathcal{H}} \sum_{y'\in\mathcal{Y}} p_{\mathcal{H}}(h')p_{\mathcal{Y}}(y')J(\check{b},h',y';\beta) \quad (7)$$

for post-decision states $\check{b} \in \mathcal{B}$. The post-decision state $\check{b}^t$ in slot t is the resulting queue state *after* the control decisions are made. Hence, we have the queue dynamics as $b^{t+1} = \check{b}^t \triangleq [b^t - a^t]^+ + r^t$. Using (6) and (7), π_1^* can be computed as the solution of the following problem:

$$\arg\max_{\substack{a:a\leq b\\ r:r\leq y}} \Big\{ f_b(r) - f_c(b) - \beta c(h,a) + J_{\rm dec}(b-a+r;\beta)\Big\}. \quad (8)$$

As we will see, studying the structural properties of the optimal policy using (8) and $J_{\rm dec}(\check{b};\beta)$ is easier than using (6) and $J(b,h,y;\beta)$. Moreover, to compute π_1^*, it is sufficient to know $J_{\rm dec}(\check{b};\beta)$. In the following, we propose an online learning algorithm for $J_{\rm dec}(\check{b};\beta)$ without requiring known PDFs. Moreover, as we will see, learning $J_{\rm dec}(\check{b};\beta)$ requires less complexity and converges faster than learning the Q function as in the conventional Q-learning algorithms.

From (6) and (7), we can write the optimality functional equation on $J_{\rm dec}(\check{b};\beta)$:

$$J_{\rm dec}(\check{b};\beta) = \sum_{h'\in\mathcal{H}} \sum_{y'\in\mathcal{Y}} p_{\mathcal{H}}(h')p_{\mathcal{Y}}(y') \max_{\substack{a:a\leq \check{b}\\ r:r\leq y'}} \Big\{ f_b(r) - f_c(\check{b}) - \beta c(h',a) + J_{\rm dec}(\check{b}-a+r;\beta)\Big\} - J_{\rm dec}(\check{b}_0;\beta) \quad (9)$$

for some arbitrary but fixed state $\check{b}_0$. The structural properties of the optimal policy are now stated.

Theorem 1: Under *Model 1*, the optimal policy π_1^* of (5) has the following properties:

1. The function $J_{\rm dec}(\check{b};\beta)$ is concave decreasing with $\check{b} \in \mathcal{B}$.
2. The scheduling action $a^*(b,h,y)$ is non-decreasing with b and y.
3. The admission control action $r^*(b,h,y)$ is non-increasing with b and non-decreasing with y.
4. The actions $a^*(b,h,y)$ and $r^*(b,h,y)$ are non-decreasing with h.

PROOF: The proof is presented in Appendix B.

We can see that with the increasing buffer occupancy b, more data should be scheduled and less traffic should be admitted in order to reduce the buffer cost. When there is more arrival traffic, more data should be scheduled as such to make 'room' for new traffic to improve the throughput. The last statement in Theorem 1 holds for i.i.d. fading channels only (assumption (A1)).

B. Stochastic Approximation Based Online Learning Algorithm

To compute the optimal policy in (8), we need to compute the value function $J_{\rm dec}(\check{b};\beta)$. Using (9), $J_{\rm dec}(\check{b};\beta)$ can be computed using the sequential relative value iteration algorithm (RVIA) as follows for $t = 0, 1, \ldots$:

$$J_{\rm dec}^{t+1}(\check{b};\beta) = \sum_{h'\in\mathcal{H}} \sum_{y'\in\mathcal{Y}} p_{\mathcal{H}}(h')p_{\mathcal{Y}}(y')\Bigg(\max_{\substack{a:a\leq \check{b}\\ r:r\leq y'}} \Big\{ f_b(r) - f_c(\check{b}) - \beta c(h',a) + J_{\rm dec}^t(\check{b}-a+r;\beta)\Big\}\Bigg) - J_{\rm dec}^t(\check{b}_0;\beta) \quad (10)$$

with initial condition $J_{\rm dec}^0(\check{b};\beta) = 0$. The purpose of subtracting the scalar offset is to keep the iterations stable. Iterations (10) converge to $J_{\rm dec}(\check{b};\beta)$ satisfying (9) [18].

The iterations (10) require known PDFs to evaluate the expectation. However, the equation (10) has a nice structure such that the expectations are moved outside of the maximization, and hence, we can use online time-averaging to learn $J_{\rm dec}(\check{b};\beta)$ under unknown PDFs, i.e., it solves the MDP (5) for a fixed β. Moreover, to find the solution of (4), the multiplier β can be updated using stochastic subgradient method. The optimality and convergence results of the online learning algorithm are ensured using the results in stochastic approximation theory. Using (10), the online updating equations are as follows:

$$\beta^{t+1} = \Lambda\big[\beta^t + \varepsilon^t\big(c(h^t,a^t) - C_{\max}\big)\big] \quad (11)$$

$$J_{\rm dec}^{t+1}(\check{b};\beta^{t+1}) = (1-\phi^t)J_{\rm dec}^t(\check{b};\beta^t) + \phi^t\Bigg(\max_{\substack{a:a\leq \check{b}\\ r:r\leq y^t}} \Big\{ f_b(r) - f_c(\check{b}) - \beta^t c(h^t,a) + J_{\rm dec}^t(\check{b}-a+r;\beta^t)\Big\} - J_{\rm dec}^t(\check{b}_0;\beta^t)\Bigg) \quad (12)$$

for $\check{b} \in \mathcal{B}$ with initial conditions $J_{\rm dec}^0(\check{b};\beta^0) = 0$, and $\beta^0 > 0$. We have used the projection operator Λ to project the multiplier onto interval $[0, L]$ for sufficiently large L to ensure boundedness of the multiplier. The learning rate sequences ϕ^t and ε^t satisfy the following properties [7]:

$$\sum_{\tau=0}^{\infty}\phi^\tau = \sum_{\tau=0}^{\infty}\varepsilon^\tau = \infty; \sum_{\tau=0}^{\infty}(\phi^\tau)^2 + (\varepsilon^\tau)^2 < \infty; \lim_{\tau\to\infty}\frac{\varepsilon^\tau}{\phi^\tau} = 0. \quad (13)$$

[6] Note that after knowing Q function, the value function $J(b,h,y;\beta)$ can also be computed.

The control actions a^t and r^t in slot t are computed as:

$$\arg\max_{\substack{a:a\leq b^t\\ r:r\leq y^t}}\left\{f_b(r)-f_c(b^t)-\beta^t c(h^t,a)+J^t_{\text{dec}}(b^t-a+r;\beta^t)\right\}. \tag{14}$$

It is worth noting that in (12), we batch-update $J^{t+1}_{\text{dec}}(\check{b};\beta^{t+1})$ for all post-decision states $\check{b}\in\mathcal{B}$, not only the previously-visited state. This is possible because the traffic arrival and the channel processes are independent of the queue states [8], [9]. The equation (12) can be viewed as a stochastic estimate of its counterpart (10), and is updated based on the instantaneous traffic arrival y^t and channel h^t states without requiring known PDFs. The optimality and convergence of the proposed learning algorithm are stated next.

Theorem 2: The functions $J^t_{\text{dec}}(\check{b};\beta^t)$ for $t=0,1,\ldots$ are concave decreasing with $\check{b}$. Moreover, $\lim_{t\to\infty} J^t_{\text{dec}}(\check{b};\beta^t)=J_{\text{dec}}(\check{b};\beta^*)$; $\lim_{t\to\infty}\beta^t=\beta^*$ where β^* is the optimal Lagrange multiplier of (4).

PROOF: The proof of the decreasing concavity property of $J^t_{\text{dec}}(\check{b};\beta^t)$ follows similar line of arguments as that of Theorem 1 in Appendix B. The convergence proof based on stochastic approximation and two-timescale analysis can be adapted from the results in [7], [19] and is omitted for brevity.

The proposed online learning algorithm does not assume any specific PDFs of the system dynamics. Hence, it is very robust to channel and traffic arrival model variations. Due to batch updates, the learning process converges faster. It is mentioned in [8] that batch updates result in twice faster convergence rate than updating one state in each slot. Also, the batch updates preserve the concavity of the value functions. Hence, the computational complexity of updating the value functions in (12) involves solving convex optimization problems. The convexity preservation of the value functions can also be exploited to derive approximate learning algorithm as in [8]. Compared to Q-learning which learns the Q function with large complexity (which is approximately $|\mathcal{B}|^2\times|\mathcal{H}|\times|\mathcal{Y}|^2$ where $|.|$ denotes cardinality of a set) and slow convergence [6], [16], the proposed learning requires less complexity (which is $|\mathcal{B}|$) and converges faster. This is because Q-learning maintains a value table for each state-action pair and updates one table entry in each slot.

We can see that the primal variables and the dual Lagrange multiplier are iterated simultaneously albeit on different timescales. The latter is updated at a slower timescale than the former. As seen from the slower timescale variable, the faster timescale variables appear to be equilibrated to the optimal values corresponding to its current value. Also, as viewed from the faster timescale variables, the slower timescale variable appears to be almost constant. Such two timescales updates converge to the optimal solution of (4) [7], [19].

The last remark regards the periodic updates in the learning algorithm. In (12), updates are performed in every slot. However, even updates are carried out with updating frequency $T_0>1$ slots or more general, at random slots using the latest information (asynchronous updates), the learning algorithm also converges to the optimal solution because all the arrival and channel states are still realized infinitely many times. Moreover, it is expected that the convergence rate is slower if the multipliers are updated less frequently.

IV. Optimal Policies for Model 2: Structural Results and Online Learning Algorithm

In *Model 2*, it is assumed that in slot t, the controller does not observe the arrival state y^t when making scheduling decision. More specifically, the scheduling action a^t is determined first based on the state (b^t,h^t) and the admission control action r^t is determined after based on the state $([b^t-a^t]^+,y^t)$. This model is commonly assumed in existing works. Hence, a stationary SAC policy π_2 for (5) consists of a scheduling policy represented by a function $a:\mathcal{B}\times\mathcal{H}\to\mathbb{R}^+$ and an admission control policy represented by a function $r:\mathcal{B}\times\mathcal{Y}\to\mathbb{R}^+$. The scheduling policy specifies a^t as a function of the state (b^t,h^t), i.e., $a^t=a(b^t,h^t)\in[0,b^t]$; The admission control policy specifies r^t as a function of the state (b^t-a^t,y^t), i.e., $r^t=r(b^t-a^t,y^t)\in[0,y^t]$.

A. Post-Transmission and Post-Admission States and Corresponding State-Value Functions

Define $V(b,h;\beta)$ as the (pre-transmission) state-value function, i.e., $V(b,h;\beta)$ is the optimal value of (5) with the starting state $(b^0,h^0)=(b,h)$. The functional Bellman's optimality equation for (5) is:

$$V(b,h;\beta)=\max_{a:a\leq b}\Big\{-f_c(b)-\beta c(h,a)+\sum_{y'\in\mathcal{Y}}p_{\mathcal{Y}}(y')\Big(\max_{r:r\leq y'}\Big\{f_b(r)+\sum_{h'\in\mathcal{H}}p_{\mathcal{H}}(h')V(b-a+r,h';\beta)\Big\}\Big)-V(b_0,h_0;\beta)\Big\} \tag{15}$$

for some arbitrary but fixed state (b_0,h_0). The optimal policy π_2^* consists of the optimal solutions of the two maximizations in (15). The equation (15) is different from that in (6) reflecting the differences in the SAC models.

We now introduce two new states and their corresponding state-value functions. The post-admission state-value function $V_{\text{ad}}(\tilde{b};\beta)$ is defined as:

$$V_{\text{ad}}(\tilde{b};\beta)=\sum_{h'\in\mathcal{H}}p_{\mathcal{H}}(h')V(\tilde{b},h';\beta) \tag{16}$$

for post-admission states $\tilde{b}\in\mathcal{B}$. Hence, the post-admission state $\tilde{b}^t$ in slot t equals to the backlog state b^{t+1} in slot $t+1$. The post-transmission state-value function $V_{\text{tr}}(\hat{b};\beta)$ is defined as:

$$V_{\text{tr}}(\hat{b};\beta)=\sum_{y'\in\mathcal{Y}}p_{\mathcal{Y}}(y')\Big(\max_{r:r\leq y'}\Big\{f_b(r)+V_{\text{ad}}(\hat{b}+r;\beta)\Big\}\Big). \tag{17}$$

for post-transmission states $\hat{b}\in\mathcal{B}$. By definition, we have the queue dynamics $\hat{b}^t=[b^t-a^t]^+$, $\tilde{b}^t=\hat{b}^t+r^t$, and $b^{t+1}=\tilde{b}^t$. From (15), we also have the following relationship:

$$V_{\text{ad}}(\tilde{b};\beta)=\sum_{h'\in\mathcal{H}}p_{\mathcal{H}}(h')\max_{a:a\leq\tilde{b}}\Big\{-f_c(\tilde{b})-\beta c(h',a)+V_{\text{tr}}(\tilde{b}-a;\beta)\Big\}. \tag{18}$$

From (15), the optimal policy π_2^* is the optimal solutions of the following problems:

$$a^*(b,h)=\arg\max_{a:a\le b}\Big\{-f_c(b)-\beta c(h,a)+V_{\rm tr}(b-a;\beta)\Big\} \quad (19)$$

$$r^*(\hat{b},y)=\arg\max_{r:r\le y}\Big\{f_b(r)+V_{\rm ad}(\hat{b}+r;\beta)\Big\}. \quad (20)$$

Hence, to compute the optimal policy π_2^*, it is sufficient to know the state-value functions $V_{\rm tr}$ and $V_{\rm ad}$. In the following, we will propose a learning algorithm for these functions.

Theorem 3: Under *Model 2*, the optimal policy π_2^* of (5) has the following properties:

1. The functions $V_{\rm ad}(\tilde{b};\beta)$, and $V_{\rm tr}(\hat{b};\beta)$ are concave decreasing with $\tilde{b}$, and $\hat{b}\in\mathcal{B}$.
2. The scheduling action $a^*(b,h)$ is non-decreasing with b and non-decreasing with h.
3. The admission control action $r^*(\hat{b},y)$ is non-increasing with $\hat{b}$, non-decreasing with y, and has the following form:
$$r^*(\hat{b},y)\ =\ \min\{\bar{B},\hat{b}+y\} \quad (21)$$
where $\bar{B}$ is some threshold.

PROOF: The proof is presented in Appendix C.

Theorem 3 says that the admission control policy can be emulated using a finite buffer with size $\bar{B}$ and the queue dynamics in (1) can be written as follows for $t=0,1,\ldots$:

$$b^{t+1}=\min\Big\{\bar{B},[b^t-a^t]^++y^t\Big\}. \quad (22)$$

All arrivals y^t are admitted whenever adding y^t does not make the backlog exceed the threshold $\bar{B}$, and else r^t is equal to only that portion of the new arrivals that take backlog up to $\bar{B}$. The EECA developed in [10] prescribes that, in every slot, all new arrivals are admitted whenever the current backlog is below a predetermined threshold. Else, all new arrivals are dropped.

More insights into the scheduling policy can be obtained using convex analysis for (19). We can see that when the state h is above some (unique) 'threshold' h^* satisfying:

$$\beta\partial c(h^*,0)/\partial a=-\partial V_{\rm tr}(0;\beta)/\partial\hat{b}, \quad (23)$$

all data is scheduled when the backlog is below the threshold $b_1(h)$ (as a function of h) satisfying:

$$\beta\partial c(h,b_1(h))/\partial a=\partial c(h^*,0)/\partial a,\quad h\ge h^*. \quad (24)$$

The transmission power is smaller than the buffer cost because of favorable channel state(s). Furthermore, when the backlog b is larger than $b_1(h)$, only a portion of the backlog is scheduled, i.e., $a^*(b,h)\in(b_1(h),b)$. This is because the power becomes large due to the convex increasing property. $b_1(h)$ can be shown to be increasing with h, $h>h^*$. When h is below h^*, no traffic is scheduled when the backlog is below another threshold $b_2(h)$ satisfying:

$$\partial V_{\rm tr}(b_2(h);\beta)/\partial\hat{b}=\beta\partial c(h,0)/\partial a,\quad h<h^* \quad (25)$$

since the buffer cost is small, and the transmission power is large. When $b>b_2(h)$, a portion of the backlog is scheduled, i.e., $a^*(b,h)\ \in\ (0,b)$. Note that $b_2(h)$ can be shown to be decreasing with h. The scheduling policy is depicted in Fig. 1. Overall, the greater the backlog and/or the better the channel state, the more you transmit. Such scheduling policy possesses similar structural properties as scheduling policy without admission control [2], [3], [8]. This is not surprising since we have shown that the admission control policy can be implemented using a queue with finite buffer $\bar{B}$.

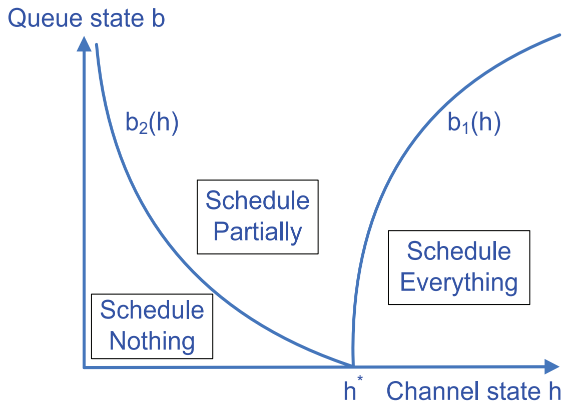

Fig. 1. Characterization of the optimal scheduling policy in *Model 2*.

B. Stochastic Approximation Based Online Learning Algorithm

To compute the optimal policy in (19) and (20), we need to compute the value functions. Using (17) and (18), the sequential RVIA equations for the value functions can be written as follows for $t=0,1,\ldots$ for $\hat{b},\tilde{b}\in\mathcal{B}$:

$$V_{\rm tr}^{t+1}(\hat{b};\beta)=\sum_{y'\in\mathcal{Y}}p_{\mathcal{Y}}(y')\Big(\max_{r:r\le y'}\Big\{f_b(r)+V_{\rm ad}^t(\hat{b}+r;\beta)\Big\}\Big) -V_{\rm tr}^t(\hat{b}_0;\beta) \quad (26)$$

$$V_{\rm ad}^{t+1}(\tilde{b};\beta)=\sum_{h'\in\mathcal{H}}p_{\mathcal{H}}(h')\Big(\max_{a:a\le\tilde{b}}\Big\{f_c(\tilde{b})-\beta c(h',a) +V_{\rm tr}^{t+1}(\tilde{b}-a;\beta)\Big\}\Big)-V_{\rm ad}^t(\tilde{b}_0;\beta) \quad (27)$$

with initial conditions $V_{\rm ad}^0(\tilde{b};\beta)=0$, $V_{\rm tr}^0(\hat{b};\beta)=0$ and $\hat{b}_0$, $\tilde{b}_0$ are arbitrary but fixed states. The iterations converge to the functions satisfying (16), (17), and (18).

Again, iterations (26) and (27) require known PDFs to evaluate the expectations. Since the expectations are outside of the maximization operators in (26)–(27), a learning algorithm can be developed using online time-averaging to learn the value functions. Also, the Lagrange multiplier in (4) can be updated using stochastic sub-gradient algorithm at a slower timescale. The updating equations are as follows for $t=0,1,\ldots$:

$$\beta^{t+1}=\Lambda\big[\beta^t+\varepsilon^t\big(c(h^t,a^t)-C_{\max}\big)\big] \quad (28)$$

$$V_{\rm tr}^{t+1}(\hat{b};\beta^{t+1})=(1-\phi^t)V_{\rm tr}^t(\hat{b};\beta^t)+\phi^t\Big(\max_{r:r\le y^t}\Big\{f_b(r) +V_{\rm ad}^t(\hat{b}+r;\beta^t)\Big\}-V_{\rm tr}^t(\hat{b}_0;\beta^t)\Big) \quad (29)$$

$$V_{\rm ad}^{t+1}(\tilde{b};\beta^{t+1})=\max_{a:a\le\tilde{b}}\Big\{-f_c(\tilde{b})-\beta^t c(h^t,a) +V_{\rm tr}^{t+1}(\tilde{b}-a;\beta^{t+1})\Big\}-V_{\rm ad}^t(\tilde{b}_0;\beta^t) \quad (30)$$

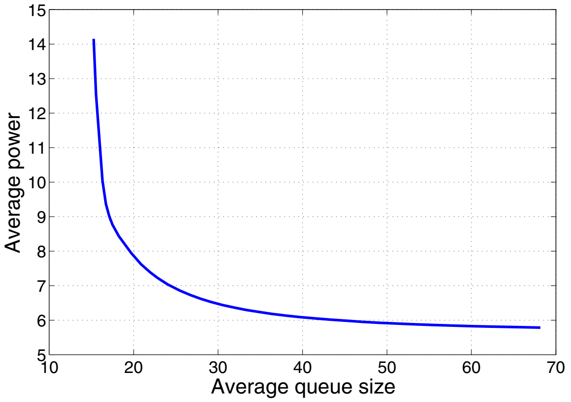

Fig. 2. Optimal power-queue size trade-off.

for $\tilde{b}, \hat{b} \in \mathcal{B}$. The initial conditions are $V_{\text{ad}}^0(\tilde{b};\beta^0) = V_{\text{tr}}^0(\hat{b};\beta^0) = 0$, $\beta^0 > 0$, and the learning rate sequences satisfy the requirement in (13). The control actions a^t and r^t in slot $t = 0, 1, \ldots$ are computed as:

$$a^t = \arg\max_{a:a\leq b^t}\Big\{-\beta^t c(h^t,a)+V_{\text{tr}}^t(b^t-a;\beta^t)\Big\} \qquad (31)$$

$$r^t = \arg\max_{r:r\leq y^t}\Big\{f_b(r)+V_{\text{ad}}^t(\hat{b}^t+r;\beta^t)\Big\}. \qquad (32)$$

Note that in (29) and (30), we also batch-update the state-value functions for all states $\hat{b}$, $\tilde{b} \in \mathcal{B}$ due to independent processes. Also, V_{ad}^{t+1} in (30) needs not to be time-averaged since time-averaging has been carried out for V_{tr}^{t+1} in the same slot. Again, we can see that the iterations (29) and (30) are updated based on the instantaneous arrival y^t and channel h^t states without requiring known PDFs. The convergence of the learning algorithm is established next.

Theorem 4: The functions $V_{\text{ad}}^t(\tilde{b};\beta^t)$ and $V_{\text{tr}}^t(\hat{b};\beta^t)$ for $t = 0, 1, \ldots$ are concave decreasing with $\tilde{b}$ and $\hat{b}$. Moreover, $\lim_{t\to\infty} V_{\text{ad}}^t(\tilde{b};\beta^t) = V_{\text{ad}}(\tilde{b};\beta^*)$, $\lim_{t\to\infty} V_{\text{tr}}^t(\hat{b};\beta^t) = V_{\text{tr}}(\hat{b};\beta^*)$; $\lim_{t\to\infty} \beta^t = \beta^*$ where β^* is the optimal Lagrange multiplier of (4).

PROOF: The proof is analogous to that of Theorem 2 and is omitted.

1) Exploiting Structural Results for Reduced Complexity Learning: The learning algorithm (28)–(30) needs to update and store the values of two value functions. Exploiting the threshold property of the optimal admission control policy (see Theorem 3), it is possible to reduce the learning (storage) complexity as follows. In each slot, we can update and store the value function $V_{\text{tr}}^{t+1}(\hat{b};\beta^{t+1})$ directly using $V_{\text{tr}}^t(\hat{b};\beta^t)$ which is required to compute the scheduling action. The admission control policy can be updated by updating the threshold $\bar{B}^{t+1}$ only. Such updates can be done as follows. Replacing J_{ad}^t in (29) using (30), we have:

$$V_{\text{tr}}^{t+1}(\hat{b};\beta^{t+1}) = (1-\phi^t)V_{\text{tr}}^t(\hat{b};\beta^t)+\phi^t\Big(\max_{r:r\leq y^t}\Big\{f_b(r)$$
$$+\max_{a:a\leq\hat{b}+r}\Big\{-f_c(\hat{b}+r)-\beta^t c(h^t,a)+V_{\text{tr}}^t(\hat{b}+r-a;\beta^t)\Big\}\Big\}$$
$$-V_{\text{tr}}^t(\hat{b}_0;\beta^t)\Big).$$

TABLE II
CHANNEL STATES USED IN THE SIMULATION

Power gain $\frac{\lvert c\rvert^2}{\sigma^2}$ regions	Representative state values h
$(0, 0.0280]$, $(0.0280, 0.0580]$, $(0.0580, 0.0960]$, $(0.0960, 0.1400]$, $(0.1400, 0.1980]$, $(0.1980, 0.2780]$ $(0.2780, 0.4160]$, $(0.4160, \infty)$	0.0131, 0.0418, 0.0753, 0.1157, 0.1661, 0.2343, 0.3407, 0.6200

The threshold $\bar{B}^{t+1}$ is then updated as such to satisfy $\partial V_{\text{ad}}^{t+1}(\bar{B}^{t+1};\beta^{t+1})/\partial\tilde{b} + \partial f_b(\bar{B}^{t+1})/\partial r = 0$ where $V_{\text{ad}}^{t+1}(\tilde{b};\beta^{t+1})$ is computed using $V_{\text{tr}}^{t+1}(\hat{b};\beta^{t+1})$ as in (30). Note that we do not need to store the values of $V_{\text{ad}}^{t+1}(\tilde{b};\beta^{t+1})$ in each slot. The threshold $\bar{B}^{t+1}$ converges to the optimal threshold $\bar{B}$ in Theorem 3 when $t \to \infty$ due to the convergence of the value functions (see Theorem 4).

V. ILLUSTRATIVE RESULTS

A. Simulation Setup

We implement the proposed learning algorithms using MATLAB. We assume that the slot duration is equal to $1/W$ where W (Hz) is the bandwidth.

To determine the channel states for illustrative purposes, the entire range $(0,\infty)$ of (normalized) power gain $\frac{\lvert c\rvert^2}{\sigma^2}$ (where σ^2 and c are, respectively, the variance of the white Gaussian noise and the channel coefficient) has been divided into 8 regions and each region is represented by a channel state value $h \in \mathcal{H}$. The 8 regions and their corresponding representative values are summarized as in Table II. The corresponding probabilities are $[1,1,2,3,3,2,1,1]/14$. Such discretization of state space of the power gains has been justified in [20], [21].

We use the exponential power function derived from the Shannon theoretic rate $c(h,a) = (2^a-1)/h$.

We assume (truncated) Poisson arrival process with an average rate 15 (bits) per slot with $y_{\min} = 0$ and $y_{\max} = 30$. The buffer size is assumed to be 1000 for implementation convenience and we observe no buffer overflow happens.

The learning rate sequences are chosen as $\phi^t = (1/t)^{.7}$ and $\varepsilon^t = (1/t)^{.85}$. The learning algorithms are run for 50000 slots for each simulation example.

To obtain the trade-off curves, we let the functions be $f_b(r) = r$ and $f_c(b) = \kappa b$ for different values of $\kappa \in (0,1)$.

B. Numerical Results

We plot in Fig. 2 the optimal power- queue size trade-off [1]. Note that the arrival state y^t cannot be observed when scheduling decision a^t is determined (Model 2). We can see that to achieve maximum throughput given an average power $C_{\max} = 6.5$, the queue size $B_{\max}$ is approximately 29 (bits). Also, the minimum power required to ensure finite queue size without traffic dropping is $C_\infty \approx 5.8$.

We are now looking at the performances of the SAC policies. Fig. 3 plots the optimal trade-off curves achieved by the proposed online learning algorithms for both models. We also plot the trade-off obtained by the ECCA in [10]. We can observe that for the same queue size, the proposed

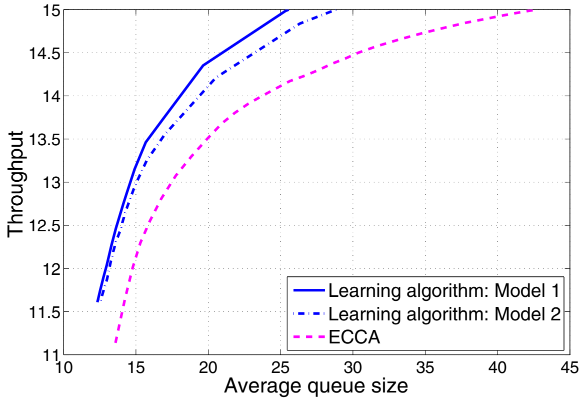

Fig. 3. Throughput- queue size trade-off for $C_{\max} > C_\infty$.

learning algorithms is able to achieve higher throughput than the ECCA. Alternatively, for the same throughput, the learning algorithms achieve smaller queue size. When the average queue size approaches $B_{\max}$ (by setting κ sufficiently small in the learning algorithms), the throughput approaches the average arrival rate, i.e., almost all the arrival is buffered. Moreover, better trade-off can be achieved when we are able to observe the realization of the arrival state (Model 1) since we can jointly optimize the scheduling and admission control decisions at the same time. The results also confirm the concavity increasing characteristic of the optimal trade-off which is analytically proved in Proposition 1.

Figs. 4 and 5 demonstrate the convergence of the proposed learning algorithm under *Model 2* for some values of κ (Theorem 4).[7] Fig. 4 shows the convergence of the Lagrange multiplier (updated using stochastic sub-gradient method) and power consumption while Fig. 5 shows the convergence of the queue size and throughput. In all cases, the learning algorithm consumes the maximum available power $C_{\max}$. We can see that the coefficient κ controls the throughput- queue size trade-off, i.e., smaller κ results in higher average queue size and higher throughput. The algorithm converges reasonably fast, especially for small values of κ. Note that for the considered systems, the duration of operation is much longer than that needed to achieve convergence. Hence, even the learning is sub-optimal at the beginning, convergence to the optimal solutions happens much earlier than the completion of the operation.

In the last experiment, we demonstrate the use of the proposed learning algorithm to stabilize the queue when the maximum power is $C_{\max} = 4.5 < C_\infty$. Fig. 6 shows the trade-off curves obtained by the proposed algorithm and the ECCA. Again, the proposed algorithm is more efficient in terms of higher throughput for a given average queue size or smaller average queue size for a given throughput. However, the performance gap is smaller compared to that in Fig. 3 for stablizable arrival process. By setting κ small, the queue size increases but finite, ensuring queue stability and at the same time, the throughput is maximized and is strictly less than the average arrival rate since traffic has to be dropped.

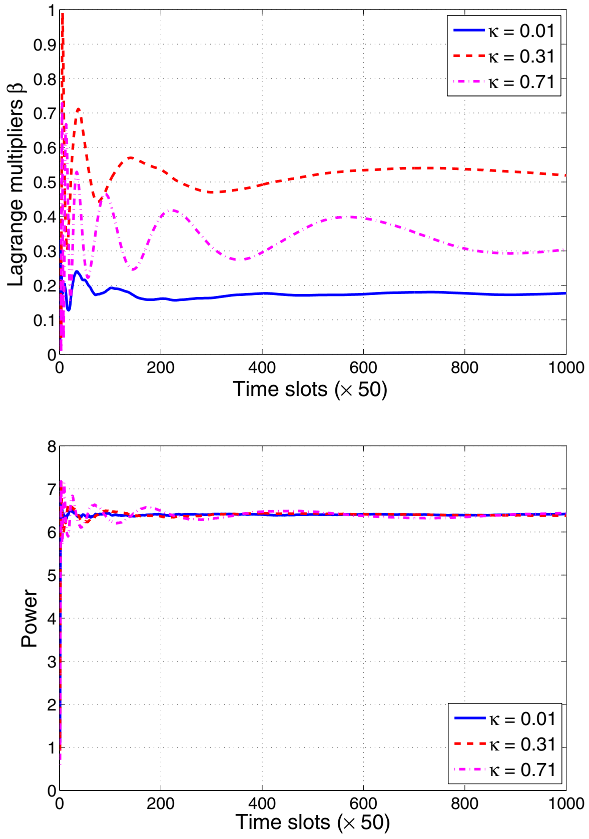

Fig. 4. Online learning convergence: Lagrange multipliers and power consumptions.

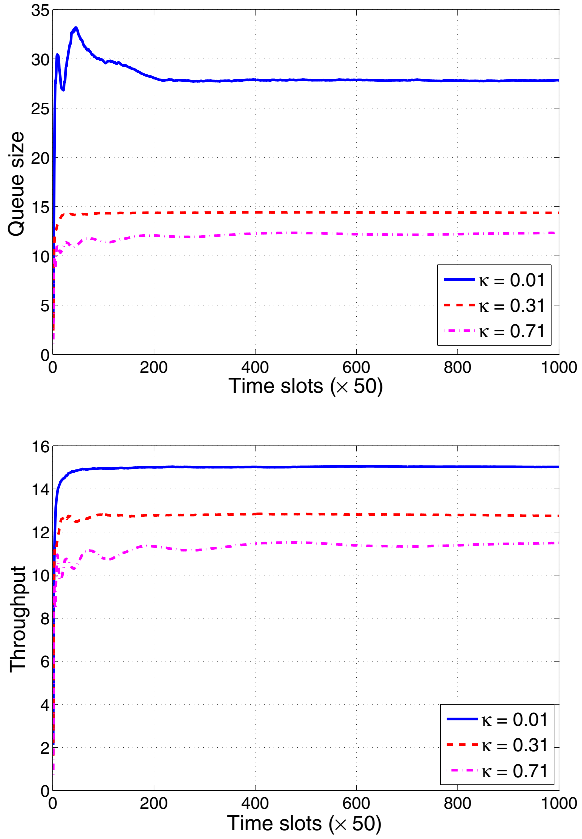

Fig. 5. Online learning convergence: Queue sizes and throughputs.

[7] We omit the simulated convergence studies for *Model 1* for brevity.

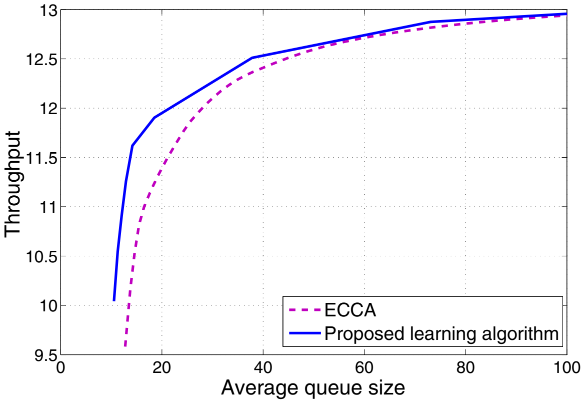

Fig. 6. Throughput- queue size trade-off for $C_{\max} < C_{\infty}$.

C. Connection with Lyapunov Optimization Based Approach

We now draw a (simple) connection between the proposed optimal learning and ECCA. For convenience, we rewrite the scheduling action in slot t as in (31) under optimal learning:

$$a^t = \arg\max_{a:a\leq b^t} \left\{-\beta^t c(h^t, a) + V_{\text{tr}}^t(b^t - a; \beta^t)\right\}. \tag{33}$$

On the other hand, ECCA minimizes the following metric to compute the scheduling action in slot t [10]:

$$\begin{aligned} a_{\text{ECCA}}^t &= \arg\max_{a:a\leq b^t} \left\{-q^t c(h^t, a) + b^t a\right\} \\ &= \arg\max_{a:a\leq b^t} \left\{-q^t c(h^t, a) - b^t(b^t - a)\right\} \end{aligned} \tag{34}$$

where a term $(b^t)^2$ is added without changing the optimal solution in (34). $\{q^t\}$ is virtual power queue state in slot t and is updated as $q^{t+1} = [q^t - c(h^t, a_{\text{ECCA}}^t)]^+ + C_{\max}$. Comparing (33) and (34), we shall have:

$$V_{\text{tr}}^t(\hat{b}; \beta^t) \approx -\alpha^t b^t \hat{b}$$

where $\alpha^t = \beta^t / q^t$ is some scaling coefficient. Hence, ECCA can be considered as an approximate learning algorithm where the value function $V_{\text{tr}}^t(\hat{b}; \beta^t)$ is approximated by a linear decreasing function with the slope $-\alpha^t b^t$. Remind that in the optimal learning, V_{tr}^t is concave decreasing. Such approximation has different effects in different traffic loading regions. For example, in the large queue size region, i.e., b^t is large, linear decreasing function is a 'good' approximation of the optimal concave decreasing function. Moreover, the two admission control policies do not have much different effects since the queue size is (effectively much) larger than the arrival state (see Fig. 6). Hence, ECCA performs well in high traffic loading region. However, in the small/medium queue size region, such approximation is coarse, which leads to a worse performance of the ECCA as seen in Fig. 3.

VI. Conclusion

This paper presents a study of the joint scheduling-admission control problem and its corresponding throughput-queue length trade-off using the Markov decision process approach and stochastic control tools. We have derived the structural properties of the optimal policies and proposed online learning algorithms for the optimal policies without requiring a-priori known probability distribution functions of the environment dynamics. The analysis and algorithm development are relied on the introduction of new state-value functions to reformulate the Bellman's dynamic programming equations. Moreover, these value functions can be learned efficiently using online time-averaging whose convergence and optimality are ensured by the results in the stochastic approximation theory. The learning algorithms require less complexity and converge faster than the conventional Q-learning algorithms. We note that in this work, we do not consider buffering non-admitted traffic. A possible future extension is to employ an additional (controller) buffer to store the non-admitted traffic. These traffic can be sent to the scheduler buffer at a later time, when the power is under-utilized. Such configuration can potentially improve the throughput of the system.

Acknowledgment: The first author would like to thank Dr. Long Le at Institut National de la Recherche Scientifique (INRS), University of Quebec for discussions and helpful comments on the paper.

Appendix: Proofs

Appendix A: Proof of Proposition 1

Fix $C_{\max}$.[8] We prove $R(B)$ is concave increasing with B. That $R(B)$ is increasing with B is obvious since more traffic can be admitted if the queue size is allowed to be larger (for the same service rate). We show that it is concave. Let B^1 and B^2 be two values of queue size with corresponding throughputs $R(B^1)$ and $R(B^2)$. We remind that $R(B)$ is the maximum throughput such that the queue size is less than or equal to B. We want to show that for any $\lambda \in [0, 1]$:

$$R(\lambda B^1 + (1-\lambda)B^2) \geq \lambda R(B^1) + (1-\lambda)R(B^2). \tag{35}$$

We will prove this using sample path arguments. Let $\{h^t(w)\}_{t=0}^{\infty}$ and $\{y^t(w)\}_{t=0}^{\infty}$ be given sample paths of the channel states and traffic arrival states. Let $\{a_1^t(w)\}_{t=0}^{\infty}$ and $\{r_1^t(w)\}_{t=0}^{\infty}$ be sequences of control actions corresponding to the policy which attains $R(B^1)$. Let $\{b_1^t(w)\}_{t=0}^{\infty}$ be the corresponding sequence of backlog states. Likewise, define $\{a_2^t(w)\}_{t=0}^{\infty}$, $\{r_2^t(w)\}_{t=0}^{\infty}$, and $\{b_2^t(w)\}_{t=0}^{\infty}$ corresponding to $R(B^2)$. Note that $a_i^t(w) \leq b_i^t(w)$ and $r_i^t(w) \leq y^t(w)$ for $i = 1, 2$ for all sample paths w and for all t. We have:

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\big\{c(h^\tau(w), a_i^\tau(w))\big\} = C_{\max}, \quad i = 1, 2 \tag{36}$$

where the expectation is taken over all sample paths. Now consider the $\lambda-$policy, a new sequences of control actions, $\{a_\lambda^t(w)\}_{t=0}^{\infty}$ and $\{r_\lambda^t(w)\}_{t=0}^{\infty}$ where for all t

$$\begin{aligned} a_\lambda^t(w) &= \lambda a_1^t(w) + (1-\lambda)a_2^t(w) && (37) \\ r_\lambda^t(w) &= \lambda r_1^t(w) + (1-\lambda)r_2^t(w). && (38) \end{aligned}$$

We show that $\lambda-$policy is a feasible policy. Let $\{b_\lambda^t(w)\}_{t=0}^{\infty}$ be the sequence of backlog states using this policy.

[8] It can be seen that the optimal policies will always achieve $C_{\max}$. Otherwise, we can increase the service rate to reduce the buffer cost or to increase the throughput leading to increased objective value.

- It can be seen that $r^t_\lambda(w) \le y^t(w)$ for all w and t.
- For scheduling sequence $\{a^t_\lambda(w)\}_{t=0}^\infty$, due to the convexity of $c(h,a)$, for each t, we have:

$$\begin{aligned} c(h^t(w), a^t_\lambda(w)) \quad \le \quad & \lambda c(h^t(w), a^t_1(w)) \\ & +(1-\lambda)c(h^t(w), a^t_2(w)) \end{aligned} \tag{39}$$

  and hence,

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} E\big\{c(h^\tau(w), a^\tau_\lambda(w))\big\} \le C_{\max}. \tag{40}$$

  Hence, the $\lambda-$policy satisfies the power constraint.
- Assume at time $t=0$, $b^0_\lambda(w) = b^0_1(w) = b^0_2(w) = 0$ for all sample paths w. By definition, we have $b^{t+1}_i(w) = b^t_i(w) - a^t_i(w) + r^t_i(w)$ for $i = 1,2$ and $t \ge 0$. Then, using recursion, we have $b^t_\lambda(w) = \lambda b^t_1(w) + (1-\lambda)b^t_2(w)$ for all t. Consequently, we conclude that $a^t_\lambda(w) = \lambda a^t_1(w) + (1-\lambda)a^t_2(w) \le b^t_\lambda(w)$ for all t.

Hence, we conclude that $\lambda-$policy is a feasible policy.

We have the average queue size by the $\lambda-$policy:

$$B^\lambda = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} E\big\{b^\tau_\lambda(w)\big\} = \lambda B^1 + (1-\lambda)B^2. \tag{41}$$

Summing both sides of (38) and taking expectations, we have:

$$R^\lambda = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} E\big\{r^\tau_\lambda(w)\big\} = \lambda R(B^1) + (1-\lambda)R(B^2) \tag{42}$$

The $\lambda-$policy achieves average queue size $B^\lambda = \lambda B^1 + (1-\lambda)B^2$ and throughput $R^\lambda = \lambda R(B^1) + (1-\lambda)R(B^2)$. Moreover, by (40), the optimal policy with average power $C_{\max}$ can achieve the same average queue size but with higher throughput. Thus, we must have $R(\lambda B^1 + (1-\lambda)B^2) \ge \lambda R(B^1) + (1-\lambda)R(B^2)$ as desired. We conclude that $R(B)$ is concave increasing with B.

Appendix B: Proof of Theorem 1

We prove the decreasing concavity property of $J_{\mathrm{dec}}(\check{b};\beta)$ with $\check{b} \in \mathcal{B}$. Note that the monotonic property is obvious since the utility is (strictly) decreasing due to (strictly) increasing buffer cost with queue size. To prove the concavity property, we show that $J^t_{\mathrm{dec}}(\check{b};\beta)$ in the RVIA equation (10) is concave for $t = 0,1,\ldots$ and since $\lim_{t\to\infty} J^t_{\mathrm{dec}}(\check{b};\beta) = J_{\mathrm{dec}}(\check{b};\beta)$, we conclude that $J_{\mathrm{dec}}(\check{b};\beta)$ is also concave. We use induction.

By initialization $J^0_{\mathrm{dec}}(\check{b};\beta) = 0$. Using induction and supposing that $J^t_{\mathrm{dec}}(\check{b};\beta)$ is concave for some $t \ge 0$. Hence, for some fixed $h \in \mathcal{H}$, by assumptions (A3) and (A4), $f_b(r) - f_c(\check{b}) - \beta c(h,a) + J^t_{\mathrm{dec}}(\check{b} - a + r;\beta)$ is jointly concave in $(\check{b}, a, r)$ for $a \in [0, \check{b}]$ and $r \in [0, y]$. Hence,

$$\max_{a:a\le\check{b}, r:r\le y} \left\{ f_b(r) - f_c(\check{b}) - \beta c(h,a) + J^t_{\mathrm{dec}}(\check{b} - a + r;\beta) \right\}$$

is concave with $\check{b}$ because the maximum of jointly concave function is also concave. Then, from (10), we have $J^{t+1}_{\mathrm{dec}}(\check{b};\beta)$ is concave since the expectation preserves the concavity. We conclude that $J_{\mathrm{dec}}(\check{b};\beta)$ is concave decreasing with $\check{b}$.

We now prove the monotonicity of the control actions. By assumptions (A3), (A4) and the concavity of $J_{\mathrm{dec}}(\check{b};\beta)$, we have the function $f_b(r) - f_c(b) - \beta c(h,a) + J_{\mathrm{dec}}(b - a + r;\beta)$ is supermodular in (b,a) for $a \in [0,b]$ and submodular in (b,r) for $r \in [0,y]$. Then, by applying Topkis's Monotonicity Theorem (Theorems 1, 2 in [22]) to (8), the scheduling action $a^*(b,h,y)$ is non-decreasing with b and the admission control action $r^*(b,h,y)$ are non-increasing with b. Moreover, that $r^*(b,h,y)$ is non-decreasing with y is obvious since when y increases, the optimization domain $[0,y]$ for r becomes larger. To show that $a^*(b,h,y)$ is non-decreasing with y, we can use change of variable $\bar{r} = y - r$ to equivalently rewrite (8) as:

$$\arg\max_{\substack{a:a\le b\\ \bar{r}:\bar{r}\le y}} \left\{ f_b(y-\bar{r}) - f_c(b) - \beta c(h,a) + J_{\mathrm{dec}}(b - a + y - \bar{r};\beta) \right\}.$$

We can see that the new optimized function is supermodular in (y,a). Hence, by Topkis's Monotonicity Theorem, $a^*(b,h,y)$ is non-decreasing with y.

The monotonicity of the control actions with respect to h can be established using the analogous arguments. We should note the function $c(h,a)$ is supermodular in (h,a) by assumption (A4). This concludes the proof of Theorem 1.

Appendix C: Proof of Theorem 3

Most of the proofs of Theorem 3 follows analogous arguments as these of Theorem 1 using the corresponding RVIA equations. Hence, it is omitted for brevity. For the threshold type solution of the admission control policy, we note that since (20) is a constrained concave maximization problem, the optimal solution $r^*(\hat{b}, y)$ is obtained by taking the derivative and setting r^* to the local maximum found on the interval $[0,y]$ (possibly achieved at the end points). The optimal admission control action $r^*(\hat{b}, y)$ is thus given by (21) where $\bar{B}$ is uniquely defined such that $\partial V_{\mathrm{ad}}(\tilde{B};\beta)/\partial\hat{b} + \partial f_b(\bar{B})/\partial r = 0$.

REFERENCES

[1] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[2] M. Goyal, A. Kumar, and V. Sharma, "Optimal cross-layer scheduling of transmissions over a fading multiaccess channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3518–3536, Aug. 2008.

[3] M. Agarwal, V. Borkar, and A. Karandikar, "Structural properties of optimal transmission policies over a randomly varying channel," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1476–1491, July 2008.

[4] D. Djonin and V. Krishnamurthy, "Transmission control in fading channels—a constrained Markov decision process formulation with monotone randomized policies," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5069–5083, Oct. 2007.

[5] A. Fu, E. Modiano, and J. Tsitsiklis, "Optimal energy allocation for delay-constrained data transmission over a time-varying channel," in *Proc. 2003 IEEE INFOCOM*.

[6] D. Djonin and V. Krishnamurthy, "Q-learning algorithms for constrained Markov decision processes with randomized monotone policies: applications to MIMO transmission control," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2170–2181, May 2007.

[7] N. Salodkar, A. Bhorkar, A. Karandikar, and V. S. Borkar, "On-line learning algorithm for energy efficient delay constrained scheduling over fading channel," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 732–742, May 2008.

[8] F. Fu and M. van der Schaar, "Structure-aware stochastic control for transmission scheduling," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 3931–3945, Nov. 2012.

[9] N. Mastronarde and M. van der Schaar, "Fast reinforcement learning for energy efficient wireless communications," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6262–6266, Dec. 2011.

[10] L. Georgiadis, M. J. Neely, and L. Tassiulas. "Resource allocation and cross-layer control in wireless networks," *Foundations Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.

[11] M. J. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, July 2006.

[12] D. I. Shuman, M. Liu, and O. Wu, "Energy efficient transmission scheduling with strict underflow constraints," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1344–1367, May 2011.

[13] H. Huang and V. K. N. Lau, "Decentralized delay optimal control for interference networks with limited renewable energy storage," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2552–2561, May 2012.

[14] F. Fu and M. van der Schaar, "A systematic framework for dynamically optimizing multi-user video transmission," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 308–320, Apr. 2010.

[15] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[17] S. Shenker, "Fundamental design issues for the future Internet," *IEEE J. Sel. Areas Commun.,* vol. 13, no. 7, pp. 1176–1188, Sep. 1995.

[18] E. Altman, *Constrained Markov Decision Processes: Stochastic Modeling.* Chapman & Hall CRC, 1999.

[19] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[20] H. Wang and N. B. Mandayam, "A simple packet scheduling scheme for wireless data over fading channels," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1055-1059, July 2004.

[21] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

[22] R. Amir, "Supermodularity and complementarity in economics: an elementary survey," *Southern Econ. J.*, vol. 71, no. 3, pp. 636–660, 2005.

Khoa T. Phan (S'05) received the B.Sc. degree with First Class Honors from the University of New South Wales (UNSW), Sydney, NSW, Australia, in 2005, the M.Sc. degree from the University of Alberta, Edmonton, AB, Canada, in 2008, and the M.Sc. degree in electrical engineering from the California Institute of Technology (Caltech), Pasadena, CA, USA, in 2009. He was a research assistant with the Electrical Engineering Department, University of California, Los Angeles (UCLA), CA, USA, from 2009 to 2011. He is now a Ph.D. student with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada. His current research interests are mathematical foundations and the control and optimization of communication networks. He is also interested in network economics, applications of game theory, and mechanism design in communication networks.

He has been awarded several prestigious fellowships including the Australian Development Scholarship, the Alberta Ingenuity Fund Student Fellowship, the iCORE Graduate Student Award, the Atwood Fellowship, the McGill Doctoral Engineering Award, and the SR Telecom Doctoral Award.

Tho Le-Ngoc (F'97) obtained his B.Eng. (with Distinction) in electrical engineering in 1976, his M.Eng. in 1978 from McGill University, Montreal, and his Ph.D. in digital communications in 1983 from the University of Ottawa, Canada.

From 1977–1982, he was with Spar Aerospace Limited and was involved in the development and design of satellite communications systems. From 1982–1985, he was an Engineering Manager of the radio group in the Department of Development Engineering of SRTelecom Inc., where he developed the new point-to-multipoint DA-TDMA/TDM Subscriber Radio System SR500. From 1985–2000, he was a Professor at the Department of Electrical and Computer Engineering, Concordia University. Since 2000, he has been with the Department of Electrical and Computer Engineering of McGill University. His research interest is in the area of broadband digital communications.

He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE), the Engineering Institute of Canada (EIC), the Canadian Academy of Engineering (CAE), and the Royal Society of Canada (RSC). He is the recipient of the 2004 Canadian Award in Telecommunications Research, and recipient of the IEEE Canada Fessenden Award (2005). He holds a Canada Research Chair (Tier I) on broadband access communications, and a Bell Canada/NSERC Industrial Research Chair on performance and resource management in broadband xDSL access networks.

Mihaela van der Schaar (F'10) is the Chancellor's Professor of Electrical Engineering at the University of California, Los Angeles. She is an IEEE Fellow, a Distinguished Lecturer of the Communications Society for 2011–2012, the Editor in Chief of the IEEE TRANSACTIONS ON MULTIMEDIA, and a member of the Editorial Board of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING.

She received an NSF CAREER Award (2004), the Best Paper Award from the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2005), the Okawa Foundation Award (2006), the IBM Faculty Award (2005, 2007, 2008), the Most Cited Paper Award from the *EURASIP: Image Communications Journal* (2006), the Gamenets Conference Best Paper Award (2011), and the 2011 IEEE Circuits and Systems Society Darlington Award for Best Paper. She has received three ISO awards for her contributions to MPEG video compression and streaming international standardization activities, and holds 33 granted US patents.

Fangwen Fu (A'11) received the bachelor's and master's degrees in electrical and electronics engineering from Tsinghua University, Beijing, China, in 2002 and 2005, respectively, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles, in 2010.

He currently works with Intel, Folsom, CA, as a Media Architect. He worked as an Intern with the IBM J. Watson Research Center, Yorktown Heights, NY, and with DOCOMO USA Labs, Palo Alto, CA. He was selected by IBM Research as one of the 12 top Ph.D. students to participate in the 2008 Watson Emerging Leaders in Multimedia Workshop in 2008. His research interests include wireless multimedia streaming, resource management for networks and systems, stochastic optimization, applied game theory, video coding, processing, and analysis.

Dr. Fu received the Dimitris Chorafas Foundation Award in 2009.