Resource Allocation over Network Dynamics without Timescale Separation

Alexandre Proutiere[†], Yung Yi[‡], Tian Lan[∗], and Mung Chiang[∗]

***Abstract*—We consider a widely applicable model of resource allocation where two sequences of events are coupled: on a continuous time axis (t), network dynamics evolve over time. On a discrete time axis $[t]$, certain control laws update resource allocation variables according to some proposed algorithm. The algorithmic updates, together with exogenous events out of the algorithm's control, change the network dynamics, which in turn changes the trajectory of the algorithm, thus forming a loop that couples the two sequences of events. In between the algorithmic updates at $[t-1]$ and $[t]$, the network dynamics continue to evolve randomly as influenced by the previous variable settings at time $[t-1]$. The standard way used to avoid the subsequent analytic difficulty is to assume the separation of timescales, which in turn unrealistically requires either slow network dynamics or high complexity algorithms. In this paper, we develop an approach that does not require separation of timescales. It is based on the use of stochastic approximation algorithms with *continuous-time controlled Markov noise*. We prove convergence of these algorithms without assuming timescale separation. This approach is applied to develop simple algorithms that solve the problem of utility-optimal random access in multi-channel, multi-radio wireless networks.**

I. Introduction

In many resource allocation problems in wireless networks, there are two sequences of events coupled together. First, on a continuous time axis (t), network dynamics evolve over time. Such dynamics could be service rate, channel state, buffer size, network topology, etc. Then, on a discrete time axis $[t]$, certain control laws update resource allocation variables according to some proposed algorithm. These variables could be contention probabilities, channel holding times, transmit powers, routes, source rates, etc. Algorithmic updates, together with exogenous events out of the algorithm's control, change the network dynamics, which in turn changes the trajectory of the algorithm, thus forming a loop that couples the two sequences of events. Examples of such systems include:

(a) Adaptive random back-off CSMA systems, where users adapt the mean of their contention window periodically, depending on the level of their buffer. The continuous time network dynamics include those of users' buffers and activities, which in turn depend on their periodic contention window updates.

(b) Systems with power control over fading channel, where transmitters adapt their powers periodically, depending on the average SINR observed between two power updates. The network dynamics here are those of the SINR's on the various links driven by the constantly evolving fading gains and by the transmit powers.

In between the algorithmic updates at $[t-1]$ and $[t]$, the network dynamics continue to evolve randomly as influenced by the previous variable settings at time $[t-1]$. This turns out to introduce substantial difficulty in these systems. The standard way used to avoid this issue is to assume the separation of timescales, i.e., that the network dynamics are either much slower or much faster than the algorithm update frequency. In the former case, either the network condition is only slowly varying, which limits the applicability of the model, or the algorithm updates very frequently, thus carrying the cost of high communication complexity, if each update involves message passing, or high computation complexity otherwise. In the latter case, the algorithm is assumed to see an averaged network behaviour, i.e., between two algorithm updates, the network dynamics have time to converge to some equilibrium. However, most resource allocation algorithms, especially those based on convex optimization, are iterative and asymptotically convergent. Assuming timescale separation, and in particular slow network dynamics, means that the algorithm achieves optimality instantaneously. Yet it is often impossible to achieve exact optimality in finite time. Even for some target suboptimality gap, instantaneously achieving it is impractical in real systems.

Throughout this paper, we do *not* assume timescale separation. Instead we take the natural and general framework where the network dynamics evolve continuously while the resource allocation algorithm updates on discrete periods. The algorithm does not need to achieve optimality in each update, nor do the network dynamics have to converge between two algorithm updates. Hence we consider a "lazy" and simple resource allocation algorithm under realistic constraints of complexity. Nonetheless, in Section II, we prove convergence of the above system under mild sufficient conditions. We only assume that the network dynamics can be modelled as a continuous time Markov process, whose generator evolves in time as it depends on the parameters updated in the discrete time by the proposed algorithm. The convergence result resembles those obtained in the stochastic approximation literature [1], [2]. In particular, our algorithm can be interpreted as a *stochastic approximation algorithm with controlled Markov "noise"* as considered by Borkar in [3], except for a subtle and important difference that widdens the applicability of the theory: here the "noise" is allowed to evolve in continuous

†: Microsoft Research, Cambridge, UK. ‡: Department of Electrical Engineering, KAIST, South Korea. ∗: Department of Electrical Engineering, Princeton University, USA. This work has been supported in part by PECASE N00014-09-0449, ONR N00014-07-1-0864, AFOSR FA9550-09-1-0643, FA9550-09-1-0134, and by the IITA (Institute of Information Technology Advancement) (IITA-2009-C1090-0902-0037).

time rather than just in discrete time.

The main theorem in Section II is mathematical in nature, yet its applications are widespread due to its generality. Due to space restriction, we limit our application focus to one of the central problems in distributed resource allocation in wireless networks. In Section III, we apply the framework to address the problem of utility-optimal scheduling in multi-channel multi-radio wireless networks. Based on recent advances ideas on adaptive CSMA algorithms [4], [5], [6], [7], we develop a simple CSMA-based distributed scheduling scheme that does not require any message passing to achieve utility-optimality.

II. Stochastic Approximation with Controlled Continuous-time Markov Noise

Stochastic approximation algorithms are discrete-time stochastic processes whose general form can be written as

$$\forall n \in \mathbb{N}, \quad x_{n+1} = x_n + a_n \xi_n, \tag{1}$$

where x_n is the *system state* at step n, a_n refers to as the *step-size*, and ξ_n is a random variable representing the *observation* during step n to update the system state at the next step. Here we consider very general algorithms where the system state x_n controls the transition rates of a continuous-time Markov process, and where the observation ξ_n actually depends on the behaviour of the latter process during step n. Such algorithms are referred to as stochastic approximation algorithms with *controlled continuous-time Markov noise*. We assume that the system state x_n is in $\mathbb{R}^L$.

A. Algorithms and assumptions

Consider the algorithm described by (1) with for all $n \in \mathbb{N}$,

$$\xi_n = h(x_n, Y_n), \text{ and } Y_n = \int_n^{n+1} f(m(t))dt. \tag{2}$$

$(m(t), t \geq 0)$ is a stochastic process with values in a finite set $\mathcal{M}$, $f : \mathcal{M} \to \mathbb{R}^K$ is an arbitrary mapping, and $h : \mathbb{R}^L \times \mathbb{R}^K \to \mathbb{R}^L$ is a bounded continuous function, Lipschitz in the first variable, uniformly over the second variable. More precisely, when $t \in [n, n+1)$, the process $m(t)$ evolves as a continuous-time Markov process $m^{x_n}(t)$ of generator G^{x_n}. For left-handed continuity, we impose that $m(n) = \lim_{t\to n, t<n} m(t)$. We assume that for any $z \in \mathbb{R}^L$, the Markov process with generator G^z is irreducible and ergodic with stationary distribution π^z, that the application $z \mapsto G^z$ is continuous and that $z \mapsto \pi^z$ is Lipschitz continuous. In the following, for all $z \in \mathbb{R}^L$, $\zeta^z(dy)$ denotes the stationary distribution of $\int_0^1 f(m^z(t))dt$, where $m^z(\cdot)$ is a Markov process of generator G^z.

We further assume that x_n remains bounded, which can be imposed by projecting (1) to a bounded subset of $\mathbb{R}^L$ (see [1]). Finally, we make the usual following assumption on the step-sizes: a_n is a decreasing sequence of positive real numbers such that $a_n \sim c/n$ as $n \to \infty$, so that $\sum_n a_n = \infty$ and $\sum_n a_n^2 < \infty$.

Intuitively, we expect that due to the decreasing step-sizes, the speed of variations of the system state decreases and tends to 0 when time grows. As a consequence, the dynamics of $m(t)$ are close to those of a Markov process with fixed generator (as if the system state was frozen), and has time to converge to its ergodic behaviour. Hence, when time grows large, we expect that the system behaves as if the observation was averaged, i.e., as if in (1), we could replace $\xi_n = h(x_n, Y_n)$ by $\int_y \zeta^{x_n}(dy)h(x_n, y)$. We formalize this intuition below.

B. Convergence analysis

Define $t(n) = \sum_{i=0}^{n-1} a_n$. To conduct the convergence analysis of the algorithm, we use a continuous-time interpolation of the system state. Define $\bar{x}$ as: for all $n \in \mathbb{N}$, for all $t \in [t(n), t(n+1))$,

$$\bar{x}(t) = x_n + (x_{n+1} - x_n) \times \frac{t - t(n)}{t(n+1) - t(n)}.$$

The following theorem provides the convergence analysis of the algorithm defined by (1)-(2).

Theorem 1: Let $T > 0$, and denote by $\tilde{x}^s(\cdot)$ the solution on $[s, s+T]$ of the following ordinary differential equation (ode):

$$\dot{x} = \int_y \zeta^{x(t)}(dy)h(x(t), f(y)), \quad \tilde{x}^s(s) = \bar{x}(s). \tag{3}$$

We have almost surely,

$$\lim_{s\to\infty} \sup_{t\in[s,s+T]} |\bar{x}(t) - \tilde{x}^s(t)| = 0. \tag{4}$$

As we expected, the theorem states that when time grows, the dynamics of the underlying continuous-time process $m(t)$ are averaged. A usefull observation is that when the ode

$$\dot{x} = \int_y \zeta^{x(t)}(dy)h(x(t), f(y))$$

represents the evolution of a stable dynamical system, with unique fixed point $x^\star$, then we deduce from Theorem 1 that almost surely: $x_n \to x^\star$ when $n \to \infty$.

It is worth noting that we could also consider constant step-sizes $a_n = a$ in the algorithm, and study its convergence. The difference is that we would obtain *weak* convergence only; for example, we would have that for a very small, x_n is close to $x^\star$ for large n with high probability.

III. Utility-optimal distributed scheduling in multi-channel wireless systems

We apply stochastic approximation methods to develop utility-optimal random access in wireless networks with multiple channels and multiple radios. The design of efficient scheduling schemes in such networks is notoriously challenging, even with a centralized scheduler, see [8], [9]. In fact it resembles NP-hard graph-coloring problems. Refer to [10] for a survey on multi-channel networks. Scheduling in a distributed manner is even harder, and all existing solutions require the use of message passing procedures that can be heavy, and offer only partial performance guarantees, see [11], [12]. It has been recently suggested (see [4], [5], [13], [7]) that in single-channel networks, CSMA-based random access protocols could be modified so as to achieve high efficiency.

The application of these ideas to multi-channel multi-radio systems is non-trivial, requiring careful treatment of the use of the various channels and radios.

A. Network model and objectives

Network model. The network consists in a set $\mathcal{V}$ of V nodes and a set $\mathcal{L}$ of L links. Denote by $s(l) \in \mathcal{V}$ and by $d(l) \in \mathcal{V}$ the transmitter and the receiver corresponding to link l. We also use the notation $v \in l$ if either $v = s(l)$ or $v = d(l)$. Node v has $c_v (> 0)$ radio interfaces or *radios*. On each link, data transmissions can be handled on any channel of a set $\mathcal{C}$ of C channels. These channels are assumed to be orthogonal in the sense that two transmissions on different links and different channels do not interfere. We model interference by a symmetric boolean matrix $A \in \{0,1\}^{L\times L}$, where $A_{kl} = 1$ if link k interferes link l when transmitting on the same channel, and $A_{kl} = 0$ otherwise. A node uses a radio interface to transmit or receive data on a given channel. Denote by R_{cl} the rate at which $s(l)$ can send data to $d(l)$ on channel c.

Feasible schedules and rate region. Interference and the limited number of radios at each node impose some constraints on the set of possible simultaneous and successful transmissions on the various links and channels. We capture these constraints with the notion of schedule. A schedule $m \in \{0,1\}^{C\times L}$ represents the activities of various links on different channels: $m_{cl} = 1$ if and only if link l is active on channel c (i.e., $s(l)$ is transmitting on channel c). A schedule m is *feasible* if all involved transmissions are successful, i.e., if for all $k, l \in \mathcal{L}$ and all $v \in \mathcal{V}$,

$$(m_{ck} = 1 = m_{cl}) \Rightarrow (A_{kl} = 0) \qquad \text{(Interference constraint)}$$

$$\sum_{l\in\mathcal{L}:v\in l}\sum_{c\in\mathcal{C}} m_{cl} \le c_v \qquad \text{(Radio interface constraint)}$$

We define by $\mathcal{M} \subset \{0,1\}^{C\times L}$ the set of the M feasible schedules. We are now ready to define the *rate region* Γ as the set of achievable long-term throughputs $\boldsymbol{\gamma} = (\gamma_l, l \in \mathcal{L})$ on the various links:

$$\Gamma = \Big\{ \boldsymbol{\gamma} : \exists \boldsymbol{\pi} \in [0,1]^M, \sum_{m\in\mathcal{M}} \pi_m = 1, \\ \forall l \in \mathcal{L}, \gamma_l \le \sum_{m\in\mathcal{M}} \pi_m \sum_{c\in\mathcal{C}} m_{cl} R_{cl} \Big\}. \quad (5)$$

In the above expression, π_m may be interpreted as the fraction of time the schedule m is activated.

Maximizing network utility. When the transmitters are saturated (i.e., they always have packets to send), the objective is to design a scheduling algorithm maximizing the network-wide utility. Specifically, let $U : \mathbb{R}^+ \to \mathbb{R}$ be an increasing, strictly concave, differentiable objective function. We wish to design an algorithm solving the following optimization problem:

$$\max \quad \Sigma_{l\in\mathcal{L}} U(\gamma_l), \quad \text{s.t.} \quad \boldsymbol{\gamma} \in \Gamma. \quad (6)$$

We denote by $\boldsymbol{\gamma}^\star = (\gamma_l^\star, l \in \mathcal{L})$ the optimizer of (6). Note that the network is assumed to handle single-hop data connections. However, the proposed framework can be extended to handle multi-hop connections (using classical back-pressure ideas).

B. Multi-channel CSMA algorithms

CSMA-based multi-access random back-off protocols are the most popular distributed protocols to share radio resources in wireless networks. One of the major challenges in extending CSMA protocols to multi-channel systems is channel coordination: before initiating a transmission on a given channel, a transmitter has to make sure that the corresponding receiver is actually ready to receive data on this channel using one of its radios. There have been many proposals to solve this issue. We assume hereafter that transmitters and receivers are coordinated.

Multi-channel CSMA. We propose the following extension of random back-off CSMA protocols to the case of multi-channel systems. The transmitter of link l has C independent Poisson clocks, ticking at rates λ_{cl}, $c \in \mathcal{C}$. When a clock c ticks, if the transmitter does have an available radio or if it is already transmitting or receiving on channel c, it does not do anything. Otherwise, it senses channel c, and checks whether the receiver has an available radio. If the channel is idle and if the receiver can receive data, it starts a transmission on channel c, and keeps the channel for an exponentially distributed period of time of average μ_{cl}. Define $\lambda_{\cdot l} = (\lambda_{cl}, c \in \mathcal{C})$ and $\mu_{\cdot l} = (\mu_{cl}, c \in \mathcal{C})$, and denote by CSMA($\lambda_{\cdot l}, \mu_{\cdot l}$) the above access protocol. We also introduce $\boldsymbol{\lambda} = (\lambda_{\cdot l}, l \in \mathcal{L})$ and $\boldsymbol{\mu} = (\mu_{\cdot l}, l \in \mathcal{L})$. When each link l runs CSMA($\lambda_{\cdot l}, \mu_{\cdot l}$), the network dynamics and performance can be analyzed using the theory of reversible Markov chains. More precisely, we have:

Proposition 1: Let $m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t)$ be the active schedule at time t. Then $(m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \ge 0)$ is a continuous-time reversible Markov chain whose stationary distribution $\boldsymbol{\pi}^{\boldsymbol{\lambda},\boldsymbol{\mu}}$ is given by

$$\forall m \in \mathcal{M}, \quad \pi_m^{\boldsymbol{\lambda},\boldsymbol{\mu}} = \frac{\prod_{l\in\mathcal{L},c\in\mathcal{C}} (\lambda_{cl}\mu_{cl})^{m_{cl}}}{\sum_{n\in\mathcal{M}} \prod_{l\in\mathcal{L},c\in\mathcal{C}} (\lambda_{cl}\mu_{cl})^{n_{cl}}},$$

where by convention $\prod_{l\in\emptyset}(\cdot) = 1$. Moreover, the link throughputs are given by

$$\forall l \in \mathcal{L}, \quad \gamma_l^{\boldsymbol{\lambda},\boldsymbol{\mu}} = \sum_{m\in\mathcal{M}} \pi_m^{\boldsymbol{\lambda},\boldsymbol{\mu}} \sum_{c\in\mathcal{C}} m_{cl} R_{cl}.$$

Proof. To prove the above result, we first consider the *free* process $(f^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \ge 0)$ with values in $\mathbb{N}^{C\times L}$. This process is obtained assuming that link l initiates transmissions on channel c of exponentially distributed durations with mean μ_{cl} according to a Poisson process of intensity λ_{cl} without accounting for the interference and radio interface constraints. $(f^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \ge 0)$ then represents the user populations in $C \times L$ independent $M/M/\infty$ queues, and hence is a continuous-time reversible Markov chain whose stationary distribution is proportional to $\boldsymbol{\xi}^{\boldsymbol{\lambda},\boldsymbol{\mu}}$, where

$$\forall m \in \mathbb{N}^{C\times L}, \quad \xi_m^{\boldsymbol{\lambda},\boldsymbol{\mu}} = \prod_{l\in\mathcal{L},c\in\mathcal{C}} (\lambda_{cl}\mu_{cl})^{m_{cl}}.$$

Now $(m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \ge 0)$ is obtained from $(f^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \ge 0)$ just truncating the state space to $\mathcal{M}$. It is then well-known from the classic theory of reversible processes (see [14]) that $(m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \ge 0)$ is also a continuous-time reversible

Markov chain whose invariant measures coincide with those of $(f^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \geq 0)$, and the proposition follows. □

C. Distributed utility-optimal scheduling schemes

Algorithm description. In the previous subsection, we have proposed multi-channel CSMA protocols whose parameters $(\lambda_{cl}, \mu_{cl}, c \in \mathcal{L})$ for link l are fixed. Next, we propose an algorithm that dynamically adapts these parameters so as to approximately solve the utility-maximization problem (6). This algorithm in turn is a stochastic approximation algorithm with controlled Markov noise. Time is divided into *frames* of fixed durations, and the transmitters of each link update their CSMA parameters (i.e., λ_{cl}, μ_{cl}) at the beginning of each frame. To do so, they maintain a virtual queue, denoted by $q_l[n]$ in frame n, for link l. The algorithm operates as follows:

UO-MC-CSMA (Utility-Optimal Multi-Channel CSMA)

1) During frame n, the transmitter of link l runs CSMA($\lambda_{\cdot l}[n], \mu_{\cdot l}[n]$), and records the sum $S_l[n]$ of the services received during this frame over all channels;
2) At the end of frame n, it updates its virtual queue according to

$$q_l[n+1] = \Big[q_l[n] + \frac{a_n}{W'(q_l[n])}\Big(U'^{-1}\Big(\frac{W(q_l[n])}{V}\Big) - S_l[n]\Big)\Big]_{q^{\min}}^{q^{\max}},$$

and sets the $\lambda_{cl}[n+1]$'s and $\mu_{cl}[n+1]$'s such that their products are equal to $\exp\{R_{cl}W(q_l[n+1])\}$.

In the above algorithm, $a : \mathbb{N} \to \mathbb{R}$ is a decreasing step size function satisfying $\sum_n a_n = \infty$ and $\sum_n a_n^2 < \infty$; $W : \mathbb{R}^+ \to \mathbb{R}^+$ is a strictly increasing and continuously differentiable function, termed the *weight function*; V, $q^{\min}$, $q^{\max}(> q^{\min})$ are positive parameters, and $[\cdot]_c^d \equiv \max(d, \min(c, \cdot))$. As shown later on, V controls the accuracy of the algorithm.

The UO-MC-CSMA algorithm is a stochastic approximation algorithm with controlled continuous-time Markov noise as considered in Section II. The equivalence is obtained as follows. $x_n \equiv \boldsymbol{q}[n] \in \mathbb{R}^L$ represents the virtual queues; $Y_n \equiv \boldsymbol{S}[n] \in \mathbb{R}^L$ represents the service received on each link in frame n (we have $K \equiv L$); $m(t)$ is the process recording the active schedule at time t under the algorithm, and is obtained in frame n as the Markov process $m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t)$ where for all l and c, λ_{cl} and μ_{cl} are such that their product is $\exp(R_{cl}W(q_l[n])$; $\boldsymbol{S}[n] = (S_1[n], \ldots, S_L[n])$ with for all l,

$$S_l[n] = \int_n^{n+1} \sum_{c\in\mathcal{C}} m_{cl}(t) R_{cl} dt, \quad \forall l,$$

hence the function $f : \mathcal{M} \to \mathbb{R}^L$ is given by: for all $m \in \mathcal{M}$, $f_l(m) \equiv \sum_c m_{cl}R_{cl}$; finally, the function $h : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}^L$ is given by: for all $\boldsymbol{q}, \boldsymbol{y} \in \mathbb{R}^L \times \mathbb{R}^L$,

$$h_l(\boldsymbol{q}, \boldsymbol{y}) \equiv \frac{1}{W'(q_l)}(U'^{-1}(W(q_l)/V) - y_l).$$

For any vector $\boldsymbol{q} \in \mathbb{R}^L$, we denote by $\pi^{\boldsymbol{q}}$ the distribution on $\mathcal{M}$ resulting from the dynamics of the CSMA($\lambda_{\cdot l}, \mu_{\cdot l}$) algorithms, where for all $l \in \mathcal{L}$ and all $c \in \mathcal{C}$, $\lambda_{cl}\mu_{cl} = R_{cl}\exp(W(q_l))$. In other words,

$$\forall m \in \mathcal{M}, \quad \pi_m^{\boldsymbol{q}} = \frac{\exp(\sum_{l\in\mathcal{L},c\in\mathcal{C}} m_{cl}R_{cl}W(q_l))}{\sum_{n\in\mathcal{M}}\exp(\sum_{l\in\mathcal{L},c\in\mathcal{C}} n_{cl}R_{cl}W(q_l))}. \quad (7)$$

We can now easily verify that the assumptions made in Section II are satisfied. First note that in view of the regularity of functions W and U, h is a bounded Lipschitz continuous function. Then it is clear that the generator of $m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t)$ is a continuous function of $\boldsymbol{q}$, and that $\boldsymbol{q} \to \pi^{\boldsymbol{q}}$ is Lipschitz continuous.

Convergence and Optimality. We now analyze the convergence and optimality of UO-MC-CSMA. For any link l, we define $\gamma_l[n] = (\sum_{i=0}^{n-1} S_l[i])/n$ the throughput achieved by link l up to frame n. To prove the convergence and optimality of UO-MC-CSMA, we will need the following assumption.

(A1) If $\boldsymbol{q}^0 \in \mathbb{R}_+^L$ solves, for all $l \in \mathcal{L}$, $W(q_l^0) = VU'(\sum_m \pi_m^{\boldsymbol{q}^0} \sum_c m_{cl}R_{cl})$, then $q^{\min} \leq q_l^0 \leq q^{\max}$, for all $l \in \mathcal{L}$.

For example, if the utility function U is such that $U'(0) < +\infty$, then (A1) is satisfied when $q^{\min} \leq W^{-1}(VU'(CR_{\max}))$ and $q^{\max} \geq W^{-1}(VU'(0))$, where $R_{\max} = \max_{c,l} R_{cl}$. The next theorem states the convergence of UO-MC-CSMA towards a point that is arbitrarily close to the utility-optimizer.

Theorem 2: Under (A1), for any initial condition $\boldsymbol{q}[0]$, UO-MC-CSMA converges in the following sense:

$$\lim_{n\to\infty} \boldsymbol{q}[n] = \boldsymbol{q}_\star \text{ and } \lim_{n\to\infty} \boldsymbol{\gamma}[n] = \boldsymbol{\gamma}_\star, \text{ almost surely},$$

where $\gamma_\star$ and $\boldsymbol{q}_\star$ are such that $(\gamma_\star, \boldsymbol{\pi}^{q_\star})$ is the solution of the following convex optimization problem (over $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$):

$$\begin{aligned} \max \quad & V\sum_{l\in\mathcal{L}} U(\gamma_l) - \sum_{m\in\mathcal{N}} \pi_m(\log \pi_m - 1) \\ \text{s.t.} \quad & \gamma_l \leq \sum_{m\in\mathcal{M}} \pi_m \sum_{c\in\mathcal{C}} m_{cl}R_{cl}, \quad \Sigma_{m\in\mathcal{M}}\pi_m = 1. \end{aligned} \quad (8)$$

Furthermore UO-CSMA approximately solves (6) as

$$\Big|\sum_{l\in\mathcal{L}} \big(U(\gamma_{\star,l}) - U(\gamma_l^\star)\big)\Big| \leq \frac{\log(M)+1}{V}. \quad (9)$$

Proof. Step 1. Averaging We first use the analysis of Section II to show that in the algorithm we can average the received services $\boldsymbol{S}[n]$. Remark that if $\zeta^{\boldsymbol{q}}(dy)$ is the stationary distribution of $\boldsymbol{S}[n]$ assuming that the virtual queues are fixed to

$\boldsymbol{q}$, we have by ergodicity: for all $l \in \mathcal{L}$, almost surely,

$$\begin{aligned}
&\int_{\boldsymbol{y}} \zeta^{\boldsymbol{q}}(dy) h_l(\boldsymbol{q},\boldsymbol{y}) \\
&= \lim_{P\to\infty} \frac{1}{P} \sum_{n=0}^{P-1} \frac{\mathbf{1}_{\{q^{\min}\le q_l \le q^{\max}\}}}{W'(q_l)} \Big[U'^{-1}(\frac{W(q_l)}{V}) \\
&\qquad - \int_n^{n+1} \sum_c m_{cl}^{\boldsymbol{q}}(u) R_{cl} du \Big] \\
&= \frac{\mathbf{1}_{\{q^{\min}\le q_l \le q^{\max}\}}}{W'(q_l)} \Big[U'^{-1}(\frac{W(q_l)}{V}) \\
&\qquad - \lim_{P\to\infty} \frac{1}{P} \int_0^P \sum_c m_{cl}^{\boldsymbol{q}}(u) R_{cl} du \Big] \\
&= \frac{\mathbf{1}_{\{q^{\min}\le q_l \le q^{\max}\}}}{W'(q_l)} \Big[U'^{-1}(\frac{W(q_l)}{V}) - \sum_{m\in\mathcal{M}} \pi_m^{\boldsymbol{q}} \sum_c m_{cl} R_{cl} \Big].
\end{aligned}$$

Now, denote by $\bar{\boldsymbol{q}}$ the continuous interpolation of $\boldsymbol{q}[n]$ (see Section II). Fix $s > 0$. Denote by $\tilde{q}^s$ the solution of the following ode, for all $l \in \mathcal{L}$,

$$\begin{aligned}
\dot{q}_l = \Big[U'^{-1}\Big(W_l(q_l)/V\Big) - \sum_{m\in\mathcal{M}} \pi_m^{\boldsymbol{q}} \sum_c m_{cl} R_{cl} \Big] \\
\times \frac{\mathbf{1}_{\{q^{\min}\le q_l \le q^{\max}\}}}{W'(q_l)},
\end{aligned} \tag{10}$$

with $\tilde{\boldsymbol{q}}^s(s) = \bar{\boldsymbol{q}}(s)$. Then applying Theorem 1, we have that, for all $T > 0$,

$$\lim_{s\to\infty} \sup_{t\in[s,s+T]} |\bar{\boldsymbol{q}}(t) - \tilde{\boldsymbol{q}}^s(t)| = 0 \quad \text{a.s.} \tag{11}$$

Now if the ode (10) is stable and has a unique fixed point $\boldsymbol{q}_\star$, then we would also have $\lim_{n\to\infty} \boldsymbol{q}[n] = \boldsymbol{q}_\star$ a.s..

Step 2. To complete the convergence proof, we show, using a similar technique as in [4] that (10) may be interpreted as a sub-gradient algorithm (projected on a bounded interval) solving the dual of the convex problem (8). The Lagrangian of (8) is given by

$$\begin{aligned}
L(\boldsymbol{\gamma},\boldsymbol{\pi};\boldsymbol{\nu},\eta) = &\Big(\sum_{l\in\mathcal{L}} VU(\gamma_l) - \nu_l\gamma_l\Big) + \Big(\sum_{l\in\mathcal{L}} \nu_l \sum_{m\in\mathcal{M}} \pi_m \sum_c m_{cl} R_{cl} \\
&- \sum_{m\in\mathcal{M}} \pi_m(\log\pi_m - 1)\Big) - \eta\Big(\sum_{m\in\mathcal{N}} \pi_m - 1\Big).
\end{aligned}$$

Then, the Karush-Kuhn-Tucker (KKT) conditions of (8) are given by, for all $l \in \mathcal{L}$, and $m \in \mathcal{M}$,

$$VU'(\gamma_l) = \nu_l, \tag{12}$$

$$-\log\pi_m + \sum_l \nu_l \sum_c m_{cl} R_{cl} - \eta = 0, \tag{13}$$

$$\nu_l \times \Big(\gamma_l - \sum_{m\in\mathcal{M}} \pi_m \sum_c m_{cl} R_{cl}\Big) = 0, \ \ \nu_l \ge 0, \tag{14}$$

$$\eta \times \Big(\sum_{m\in\mathcal{M}} \pi_m - 1\Big) = 0, \tag{15}$$

Introduce the variables $\boldsymbol{q}$ such that for all l, $q_l = W^{-1}(\nu_l)$, and the bounds $\nu^{\min} = W(q^{\min})$, $\nu^{\max} = W(q^{\max})$. By choosing

$$\eta = \log\Big(\sum_m \exp\big(\sum_{l,c} W(q_l) m_{cl} R_{cl}\big)\Big),$$

$\pi^{\boldsymbol{q}}$ (given in (7)) solves (13) and (15). Now the sub-gradient algorithm corresponding to (12) (when accounting for (14)) is

$$\dot{\nu}_l = \Big(U'^{-1}\Big(\frac{\nu_l}{V}\Big) - \sum_{m\in\mathcal{M}} \pi_m^{\boldsymbol{q}} \sum_c m_{cl} R_{cl} \Big). \tag{16}$$

Note that (16) is equivalent to (10), provided that the $\nu_l(t)$'s remain in $[\nu^{\min}, \nu^{\max}]$. But thanks to Assumption (A1), the fixed points of (16) actually belongs to $[\nu^{\min}, \nu^{\max}]$. Finally, since (8) is a strictly convex optimization problem, (16) converges to its unique equilibrium $\boldsymbol{\nu}_\star$, and hence (10) converges to $\boldsymbol{q}_\star$ such that for all $l \in \mathcal{L}$, $W(q_{\star,l}) = \nu_{\star,l}$. Using Step 1, we conclude that almost surely, $\boldsymbol{q}[n]$ converges to $\boldsymbol{q}_\star$. The convergence of $\boldsymbol{\gamma}[n]$ to $\boldsymbol{\gamma}_\star$ follows.

To prove the inequality (9), we just note that (6) is equivalent to the following optimization problem:

$$\begin{aligned}
\max \quad & V\Sigma_{l\in\mathcal{L}} U(\gamma_l) \\
\text{s.t.} \quad & \gamma_l \le \sum_{m\in\mathcal{M}} \pi_m \sum_c m_{cl} R_{cl}, \ \ \sum_{m\in\mathcal{M}} \pi_m = 1.
\end{aligned} \tag{17}$$

Eq. (9) is obtained by comparing (8) and (17), and using the fact that the entropy $\sum_m \pi_m \log\pi_m$ is always bounded by $\log M$. The proof of Theorem 2 is completed. □

References

[1] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[2] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency (Cambridge University Press), 2008.

[3] ——, "Stochastic approximation with controlled markov noise," *Systems and control letters*, vol. 55, pp. 139–145, 2006.

[4] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proceedings of Allernton Conference*, Sep. 2008.

[5] J. Shin, D. Shah, and S. Rajagopalan, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *Proceedings of ACM Sigmetrics*, 2009.

[6] J. Lee, J. Lee, Y. Yi, S. Chong, A. Proutiere, and M. Chiang, "Implementing utility-optimal CSMA," in *Proceedings of Allerton conference*, 2009.

[7] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and V. H. Poor, "Towards utility-optimal random access without message passing," *Wireless Communications and Mobile Computing, Wiley*, to appear.

[8] M. Alicherry, R. Bhatia, and E. L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proceedings of ACM Mobicom*, 2005.

[9] M. S. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proceedings of ACM Mobicom*, 2005.

[10] P. Kyasanur, J. So, C. Chereddi, and N. Vaidya, "Multi-channel mesh networks: Challenges and protocols," *IEEE Transactions on Wireless Communications*, vol. 13-2, no. 2, pp. 30–36, 2006.

[11] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks," in *Proceedings of IEEE Infocom*, 2007.

[12] S. Merlin, N. H. Vaidya, and M. Zorzi, "Resource allocation in multi-radio multi-channel multi-hop wireless networks," in *Proceedings of IEEE Infocom*, 2008.

[13] J. Ni and R. Srikant, "Distributed csma/ca algorithms for achieving maximum throughput in wireless networks," in *Proceedings of the Information Theory and Applications Workshop*, 2009.

[14] F. Kelly, *Reversibility and Stochastic Networks*. Wiley, Chichester, 1979.