Markov Approximation for Combinatorial Network Optimization

Minghua Chen, *Senior Member, IEEE*, Soung Chang Liew, *Fellow, IEEE*, Ziyu Shao, *Member, IEEE*, and Caihong Kai

***Abstract*—Many important network design problems are fundamentally combinatorial optimization problems. A large number of such problems, however, cannot readily be tackled by distributed algorithms. The Markov approximation framework studied in this paper is a general technique for synthesizing distributed algorithms. We show that when using the log–sum–exp function to approximate the optimal value of any combinatorial problem, we end up with a solution that can be interpreted as the stationary probability distribution of a class of time-reversible Markov chains. Selected Markov chains among this class yield distributed algorithms that solve the log–sum–exp approximated combinatorial network optimization problem. By examining three applications, we illustrate that the Markov approximation technique not only provides fresh perspectives to existing distributed solutions, but also provides clues leading to the construction of new distributed algorithms in various domains with provable performance. We believe the Markov approximation techniques will find applications in many other network optimization problems.**

***Index Terms*—Combinatorial optimization, distributed algorithms, Markov approximation, time-reversible Markov chains, wireless networks.**

I. Introduction

Many important network design and resource allocation problems are combinatorial in nature. Two well-studied examples are:

1) The maximum weighted independent set (MWIS) problem of finding the independent set with the maximum weight. MWIS problem is known to be a bottleneck of the wireless utility maximization problem [1].

Manuscript received February 03, 2010; revised March 10, 2012; accepted March 23, 2013. Date of publication June 14, 2013; date of current version September 11, 2013. This work was supported in part by the China 973 Program under Project 2012CB315904, in part by the General Research Funds under Projects 411209, 411010, 411011, and 414812, and in part by the Area of Excellence Grant under Project AoE/E-02/08, all established under the University Grant Committee of the Hong Kong SAR, China, and by the China NSFC Grant under Projects 61271277 and 61202459, the Open Project of Shenzhen Key Lab of Cloud Computing Technology and Application, and two gift grants from Microsoft and Cisco.

The authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: minghua@ie.cuhk.edu.hk; soung@ie.cuhk.edu.hk; ziyu.shao@gmail.com; caihong.kai@gmail.com).

Communicated by R. D. Yates, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2013.2268923

2) The optimal neighbor selection problem in peer-to-peer streaming systems of finding the “best” set of neighbors for every peer to maximize the overall streaming rate [2].

These formulations, while elegant, often suffer from two shortcomings: (i) the optimization problem could be intractable when the network size is large (i.e., it is NP-complete); (ii) algorithms to solve the optimization problem are not amenable to distributed implementation.

This paper attempts to tackle issue (ii). Specifically, we propose a general Markov approximation technique that allows us to solve many combinatorial network optimization problems in a distributed manner. “Distributed” here means either fully distributed solution or distributed solution with small message passing overheads. This also addresses issue (i) to a certain extent because the distributed implementation often allows parallel processing by different network elements in the network. Moreover, systems running distributed algorithms, compared with those running centralized algorithms, are often more adaptable to users joining and leaving the systems (e.g., peer churn in peer-to-peer systems) and are more robust to system/network dynamics (e.g., channel fading in wireless networks).

An example of the application of Markov approximation is the design of the optimal throughput CSMA mechanism. In [3] and [4], it was shown that the throughput of links in a CSMA network can be computed from a time-reversible Markov chain. Jiang and Walrand [5] and Liu *et al.* [6] used reverse-engineering to show that CSMA solves the combinatorial MWIS problem asymptotically, off by an entropy term. With this observation, Jiang and Walrand [5] and Liu *et al.* [6] made an excellent contribution showing that a standard wireless utility maximization problem [1] can be solved by running distributed algorithms on top of CSMA, with an entropy term added to the utility function. The appearance of the entropy term is a consequence of solving the utility maximization problem on top of CSMA. It turns out that similar entropy term also arises in several other existing communication systems [7], [8].

These observations naturally lead to several interesting forward-engineering questions. What is the fundamental cause of the appearance of the entropy term in all these problems? By adding an entropy term to the objective function of a combinatorial optimization problem, can we get a distributed solution out of it? If yes, how to do so systematically?

This work answers the above questions, and advocates using the entropy term as a forward-engineering device that can provide clues to the design of new distributed algorithms for various network combinatorial problems. This expands the usefulness of the approach originally expounded in the series of work in [3]–[6] to many other domains beyond CSMA networks. In particular, this paper makes the following contributions.

1) It shows that an entropy term appears as a direct consequence of approximating the optimal value of *any* combinatorial problem using a log–sum–exp function.
2) It shows that as a result of the log–sum–exp approximation, the optimal solution can be realized by the stationary distribution of a class of time-reversible Markov chains (all with the same stationary distribution).
3) It shows that selected time-reversible Markov chains in this class yield distributed algorithms that solve the log–sum–exp approximated problem.
4) It demonstrates two new applications of Markov approximation that are of much practical interest. The first is the optimal path selection problem in multipath transmission. The second is the problem of frequency channel assignment in wireless LANs located in the vicinity of each other.

The rest of this paper is organized as follows. We first present the Markov approximation technique in Section II. In Section III, we apply the Markov approximation technique to the wireless utility maximization problem and derive solutions similar to those in [5] and [6]. The goal of Section III is to provide a new perspective to the design of an existing distributed solution [5], [6]. In Sections IV and V, our goal shifts to that of applying Markov approximation to synthesize new distributed algorithms in new problem domains. Section IV studies the optimal path selection problem in multipath transmission over wireline networks. Section V investigates the problem of frequency channel assignment to Wireless LANs. Section VI concludes this paper.

II. Markov Approximation

A. Settings

Consider a network with a set of users R, and a set of configuration $\mathcal{F}$. A network configuration $f \in \mathcal{F}$ consists of individual users using one of its local configurations. When the system operates under a configuration f, each user obtains certain performance, denoted by $x_r(f)$ $(r \in R)$.[1] The problem of maximizing the system performance, i.e., aggregate user performance, by choosing the best configuration can then be cast as the following combinatorial optimization problem[2]

$$\mathbf{MWC}: \quad \max_{f\in\mathcal{F}} \quad \sum_{r\in R} x_r(f). \tag{1}$$

It has the same optimal value as the following problem:

$$\begin{aligned}\mathbf{MWC-EQ}: \quad & \max_{\boldsymbol{p}\geq 0} \quad \sum_{f\in\mathcal{F}} p_f \sum_{r\in R} x_r(f) \\ & \text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1,\end{aligned} \tag{2}$$

where p_f is the percentage of time that the configuration f is in use. Treating $\sum_{r\in R} x_r(f)$ in (1) as the "weight" of f, the problem $\mathbf{MWC}$ is to find a maximum weighted configuration.

[1]Note $x_r(f)$ can be some direct system measurement, e.g., throughput, under configuration f, or a function of the measurement.

[2]There could be other forms of combinatorial optimization problems. In this paper, we focus on the form given in (1).

Many practical and important problems are combinatorial in nature and can be formulated in the form of (1). Some well-studied examples are listed at the very beginning of Section I, and we will study three concrete examples in Sections III–V.

For many of these problems, formulation (1) could be very challenging to solve, even in a centralized manner. For example, the MWIS problem is known to be NP-hard. In practice, it is often acceptable to solve the problem approximately provided the approximation is tight. Given a good approximation, what is perhaps more important from the engineering standpoint is whether the approximation algorithm for solving the problem can be implemented in a distributed manner. This is because systems running distributed algorithms are more robust to variations due to user and system dynamics than those running centralized algorithms.

In the following, we describe a framework, which we call Markov approximation, to approach the problem in (1). Markov approximation often leads to distributed algorithms that can be implemented in practice with limited or no message passing among users, as will be demonstrated in Sections III–V.

B. Log–Sum–Exp Approximation

To gain insights on the structure of the problem $\mathbf{MWC}$, we first define a differentiable function as follows:

$$g_\beta(\boldsymbol{x}) \triangleq \frac{1}{\beta}\log\left(\sum_{f\in\mathcal{F}}\exp\left(\beta\sum_{r\in R}x_r(f)\right)\right), \tag{3}$$

where β is a positive constant and $\boldsymbol{x} \triangleq \left[\sum_{r\in R} x_i(f), f \in \mathcal{F}\right]$. Then, we use this function to approximate the max function in (1):

$$\max_{f\in\mathcal{F}} \sum_{r\in R} x_r(f) \approx g_\beta(\boldsymbol{x}). \tag{4}$$

This approximation is known as the convex log–sum–exp approximation to the max function. The approximation gap is upper-bounded by $\frac{1}{\beta}\log|\mathcal{F}|$, as follows [9, p. 72].

Proposition 1: For a positive constant β and n nonnegative real variables $y_1, y_2, \ldots, y_n$, we have

$$\begin{aligned}\max_{i=1,2,\ldots,n} y_i &\leq \frac{1}{\beta}\log\left(\sum_{i=1}^{n}\exp(\beta y_i)\right) \\ &\leq \max_{i=1,2,\ldots,n} y_i + \frac{1}{\beta}\log n.\end{aligned} \tag{5}$$

Hence, $\max_{i=1,2,\ldots,n} y_i = \lim_{\beta\to\infty} \frac{1}{\beta}\log\left(\sum_{i=1}^{n}\exp(\beta y_i)\right)$.

We summarize two useful observations of $g_\beta(\boldsymbol{x})$ in the following theorem.

Theorem 1: For the log–sum–exp function $g_\beta(\boldsymbol{x})$, we have

1) its conjugate function[3] is given by

$$g_\beta^*(\boldsymbol{p}) = \begin{cases}\frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f, & \text{if } \boldsymbol{p} \geq 0 \text{ and } \mathbf{1}^T\boldsymbol{p} = 1;\\ \infty, & \text{otherwise,}\end{cases} \tag{6}$$

2) it is a convex and closed function; hence, the conjugate of its conjugate $g_\beta^*(\boldsymbol{p})$ is itself, i.e., $g_\beta(\boldsymbol{x}) = g_\beta^{**}(\boldsymbol{x})$.

[3]Definition of conjugate function is as follows: let $g(\boldsymbol{y})$ be a $\mathbf{R}$-value function with domain $\mathbf{dom}g \in \mathbf{R}^n$, its conjugate function is defined as $g^*(z) = \sup_{\boldsymbol{y}\in\mathbf{dom}g}(z^T\boldsymbol{y} - g(\boldsymbol{y}))$ [9].

In other words, $g_\beta(\boldsymbol{x})$ is the same as the optimal value of the following problem:

$$\max_{\boldsymbol{p}\geq 0} \sum_{f\in\mathcal{F}} p_f \sum_{r\in R} x_r(f) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \quad (7)$$
$$\text{s.t.} \sum_{f\in\mathcal{F}} p_f = 1.$$

Proof: The proof of (6) follows the exposition in [9, p. 93]. $g_\beta(\boldsymbol{x})$ is a convex function because the log–sum–exp function is a convex function [9]. Further, $g_\beta(\boldsymbol{x})$ is continuous, and its domain is a closed set, thus $g_\beta(\boldsymbol{x})$ is a closed function. Hence by [9, Sec. 3.3.2], the conjugate of its conjugate $g^*_\beta(\boldsymbol{p})$ is itself. ■

Remarks: Several observations can be made. First, by the log–sum–exp approximation in (4), we are implicitly solving an approximated version of the problem $\mathbf{MWC-EQ}$, off by an *entropy* term $-\frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f$.[4] The corresponding approximation gap is bounded by $\frac{1}{\beta}\log|\mathcal{F}|$, according to Proposition 1. We emphasize that this is a direct consequence of our approximating the max function by a log–sum–exp function in (4). Practically, we argue in this paper that adding this additional entropy term in fact opens new design space for exploration. Second, the approximation becomes exact as β approaches infinity. However, as we will see in the discussions of specific applications of Markov approximation later, there are practical constraints or overhead concerns of using large β. Third, we can derive a closed-form of the optimal solution to problem (7). Let λ be the Lagrange multiplier associated with the equality constraint in (7) and $p^*_f(\boldsymbol{x})$, $f\in\mathcal{F}$ be the optimal solution of the problem in (7). By solving the Karush–Kuhn–Tucker (KKT) conditions [9] of problem (7):

$$\sum_{r\in R} x_r(f) - \frac{1}{\beta}\log p^*_f(\boldsymbol{x}) - \frac{1}{\beta} + \lambda = 0 \ \ \forall f\in\mathcal{F}, \quad (8)$$
$$\sum_{f\in\mathcal{F}} p^*_f(\boldsymbol{x}) = 1; \quad (9)$$
$$\lambda \geq 0, \quad (10)$$

we have

$$p^*_f(\boldsymbol{x}) = \frac{\exp\left(\beta\sum_{r\in R} x_r(f)\right)}{\sum_{f'\in\mathcal{F}}\exp\left(\beta\sum_{r\in R} x_r(f')\right)} \ \forall f\in\mathcal{F}. \quad (11)$$

To this end, a useful insight is that by time-sharing among different configurations f according to their portions $p^*_f(\boldsymbol{x})$, we can solve the problem $\mathbf{MWC-EQ}$, and hence the problem $\mathbf{MWC}$, approximately.

C. Algorithm Design via Markov Chain

To proceed, the idea is to design a Markov chain with the state space being $\mathcal{F}$ and the stationary distribution being $p^*_f(\boldsymbol{x})\ \ (f\in\mathcal{F})$. If we can implement the designed Markov chain in a *distributed* manner, then as the Markov chain converges, the configurations will be time-shared according to $\boldsymbol{p^*}(\boldsymbol{x})$ and the system will stay in the "best" configurations most of the time. Consequently, the average system performance will be close to the optimal value of problem $\mathbf{MWC}$, with an optimality gap of $\frac{1}{\beta}\log|\mathcal{F}|$. The following proposition shows that it suffices to consider time-reversible Markov chain in order to achieve a product-form stationary distribution $\boldsymbol{p^*}(\boldsymbol{x})$.

[4]Under the context of CSMA scheduling, Jiang and Walrand arrive at a similar observation using a different approach. We will discuss more details when we come to CSMA utility maximization in Section III.

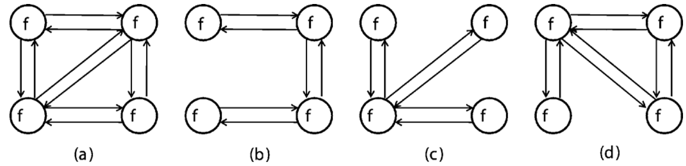

Fig. 1. Markov chains in (b), (c), (d), by adding/removing transition edge-pairs between two states in the time-reversible Markov chain in (a), are also time-reversible. All Markov chains have the same stationary distribution.

Lemma 1: For any probability distribution of the product form $p^*_f(\boldsymbol{x})$ in (11), there exists at least one continuous-time time-reversible ergodic Markov chain whose stationary distribution is $p^*_f(\boldsymbol{x})$. Further, for any continuous-time time-reversible ergodic Markov chain, its stationary distribution can be expressed in product-form $p^*_f(\boldsymbol{x})$ in (11).

The proof is relegated to Appendix-A.

To construct a time-reversible Markov chain with stationary distribution $p^*_f(\boldsymbol{x})\ \ (f\in\mathcal{F})$, we let $f\in\mathcal{F}$ be the state of the Markov chain, and $q_{f,f'}$ be the nonnegative transition rate between two states f and f'. It suffices to design $q_{f,f'}$ so that

1) the resulting Markov chain is irreducible, i.e., any two states are reachable from each other,
2) and the detailed balance equation is satisfied: for all f and f' in $\mathcal{F}$ and $f\neq f'$, $p^*_f(\boldsymbol{x})qf, f' = p^*_{f'}(\boldsymbol{x})qf', f$, i.e.,

$$\exp\left(\beta\sum_{r\in R} x_r(f)\right) q_{f,f'} = \exp\left(\beta\sum_{r\in R} x_r(f')\right) q_{f',f}.$$

We remark that the above two sufficient requirements allow a large degree of freedom in design.

First, it allows us to set the transition rates between any two states to be zero, i.e., cutting off the direct transition between them, given that they are still reachable from any other states. The modified Markov chain is still time-reversible and its stationary distribution is still $p^*_f(\boldsymbol{x})\ \ (f\in\mathcal{F})$. For example, assume the 4-states Markov chain in Fig. 1(a) is time-reversible. The "sparse" Markov chains in Fig. 1(b)–(d), modified from the "dense" one in Fig. 1(a) by adding/removing transition edge-pairs between two states, are also time-reversible. Furthermore, all Markov chains share the same stationary distribution.

Second, for two states f and f' that have direct transitions, there are many options in designing $q_{f,f'}$ and $q_{f',f}$. With α denoting a positive constant, these options include, but are not limited to, the following ones:

OPT1: Let $q_{f,f'}$ be negatively correlated to the system performance $\sum_{r\in R} x_r(f)$ under configuration f; specifically,

$$q_{f,f'} = \alpha\left[\exp\left(\beta\sum_{r\in R} x_r(f)\right)\right]^{-1}, \quad (12)$$

and $q_{f',f}$ is defined symmetrically.

OPT2: Let $q_{f,f'}$ be positively correlated to the system performance under the target configuration f'; specifically,

$$q_{f,f'} = \alpha \exp\left(\beta \sum_{r\in R} x_r\left(f'\right)\right), \tag{13}$$

and $q_{f',f}$ is defined symmetrically.

OPT3: Let $q_{f,f'}$ be positively correlated to the difference of system performance under configuration f and f'; in particular,

$$q_{f,f'} = \alpha \exp\left(\frac{1}{2}\beta \sum_{r\in R} \left(x_r\left(f'\right) - x_r\left(f\right)\right)\right), \tag{14}$$

and $q_{f',f}$ is defined symmetrically.

OPT4: Let $q_{f',f} = \alpha$, and $q_{f,f'}$ be positively correlated to the difference of system performance under configuration f and f', i.e.,

$$q_{f,f'} = \alpha \exp\left(\beta \sum_{r\in R} \left(x_r\left(f'\right) - x_r\left(f\right)\right)\right). \tag{15}$$

Design option OPT1 implies the transition rate from f to f', i.e., $q_{f,f'}$, is *independent* of the performance under target configuration f'. In contrast, $q_{f,f'}$ in OPT2 depends only on the performance of target configuration f'. Design of $q_{f,f'}$ in OPT3 combines flavors from previous two options, where the system is more likely to switch to a configuration with better performance. In practice, both OPT2 and OPT3 require the system to know the performance under target configuration f' *a priori*, or through a probing phase. Option OPT4 is similar to OPT3, but $q_{f,f'}$ and $q_{f',f}$ are no longer symmetric. As we will discuss in Section III, CSMA protocol in fact implements a Markov chain with transition rates following OPT4.

Recall that in our setting a configuration f consists of each individual user using one of its local configurations. Transitions between f and f' are done via users switching their local configurations accordingly. By users running individual continuous-time clocks and waiting for performance-dependent amounts of time before they switching their local configurations, we can design transitions to happen only between two configurations f and f' that differ by only one user's local configuration. This way, the Markov chain can be implemented in a distributed manner.

Note simulated annealing [10] also uses Markov chains for algorithm design. The difference between simulated annealing and our work is that simulated annealing in general focuses on solving the problem exactly using centralized algorithms, while we focus on designing distributed algorithms to solve the optimization problem approximately. In Sections III –VI, we present three applications of Markov approximation to illustrate its usefulness.

III. Case 1: Utility Maximization in CSMA Networks

In this section, we apply the Markov approximation technique to the wireless utility maximization problem. We derive solutions similar to those in [5] and [6]. By doing so, we wish to provide new perspectives to the design of existing distributed solutions.

A. Settings

Consider a hidden-node-free[5] CSMA wireless network, denoted by $G = (N, L)$ where N is the set of nodes and L is the set of links, each having unit capacity. Note the results can be readily extended to the case where links have heterogeneous capacities. Let its corresponding conflict graph be $G_c = (L, A)$, where A is the set of arcs in G_c. Let $\mathcal{F}$ be the set of all independent sets over G_c.

Let S be the set of all users, where a user $s \in S$ is associated with a *single* route connecting its source and destination nodes. Let $\boldsymbol{z} = [z_s, s \in S]^T$ be the vector of user rates. Let $\boldsymbol{p} = [p_f, f \in \mathcal{F}]^T$ be the vector of percentages of time an independent set is active. Let $U_s(z_s)$ be the utility function of user s upon sending at rate z_s. We assume the utility functions to be twice differentiable, increasing and strictly concave.

B. Wireless Utility Maximization Problem

Consider the following utility maximization problem over G_c:

$$\begin{aligned} \textbf{MP}: \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} & \sum_{s\in S} U_s(z_s) \\ \text{s.t.} \quad & \sum_{s:l\in s, s\in S} z_s \leq \sum_{f:l\in f} p_f \quad \forall l \in L, \\ & \sum_{f\in\mathcal{F}} p_f = 1, \end{aligned} \tag{16}$$

where $\sum_{s:l\in s, s\in S} z_s$ is the aggregate rate passing through link l, and the first set of constraints says that the incoming rate of every link cannot exceed the average link throughput.

By relaxing the first set of inequality constraints, we obtain its partial Lagrangian as follows:

$$L(\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{\lambda}) = \sum_{s\in S} U_s(z_s) - \sum_{l\in L} \lambda_l \left(\sum_{s:l\in s, s\in S} z_s - \sum_{f:l\in f} p_f\right),$$

where $\boldsymbol{\lambda} = [\lambda_l, l \in L]^T$ is the vector of Lagrange multipliers. We notice $\sum_{l\in L} \lambda_l \sum_{f:l\in f} p_f = \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l$.

Since the problem **MP** is concave and Slater's condition holds, the strong duality holds. Consequently, the optimal solution of problem **MP** can be found by solving the following problem successively in $\boldsymbol{p}$, $\boldsymbol{z}$, and $\boldsymbol{\lambda}$:

$$\begin{aligned} \min_{\boldsymbol{\lambda}\geq 0} \quad \max_{\substack{\boldsymbol{z}\geq 0 \\ \boldsymbol{p}\geq 0}} & \sum_{s\in S} U_s(z_s) - \sum_{l\in L} \lambda_l \sum_{\substack{s:l\in s, \\ s\in S}} z_s + \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l \\ \text{s.t.} & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned} \tag{17}$$

[5]In CSMA networks, two links are allowed to transmit simultaneously if they are considered to be feasible under CSMA protocol. However, CSMA protocol schedules transmissions based on carrier sensing, independent of the underlying interference model. Consequently, simultaneous transmissions allowed by CSMA may still interfere with each other, resulting in the infamous hidden-node problem. As compared to CSMA networks with hidden nodes, hidden-node-free CSMA networks are attractive not only because they are more fair, but also because their throughput analysis are more tractable. As studied in [11], a CSMA network can always be made hidden-node-free, by setting the carrier sensing threshold properly. Hence, we focus on hidden-node-free CSMA networks in our analysis.

The key challenge lies in solving the combinatorial subproblem in $\boldsymbol{p}$, which is the NP-hard MWIS problem [1]:

$$\mathbf{MWIS}: \quad \max_{\boldsymbol{p}\geq 0} \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l \quad \text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1. \tag{18}$$

The optimal value of the problem $\mathbf{MWIS}$ is given by computing the max function: $\max_{f\in\mathcal{F}} \sum_{l\in f} \lambda_l$.

C. Markov Approximation

Observing that the problem $\mathbf{MWIS}$ is a combinatorial optimization problem, we apply the Markov Approximation. First, we apply the log–sum–exp approximation

$$\max_{f\in\mathcal{F}} \sum_{l\in f} \lambda_l \approx \frac{1}{\beta} \log \left[\sum_{f\in\mathcal{F}} \exp\left(\beta \sum_{l\in f} \lambda_l\right)\right], \tag{19}$$

where β is a positive constant. According to Theorem 1, we are implicitly solving an approximated version of the problem $\mathbf{MWIS}$, off by an entropy term $-\frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f$, as follows:

$$\max_{\boldsymbol{p}\geq 0} \quad \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \quad \text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1, \tag{20}$$

and the corresponding (unique) optimal solution is

$$p_f(\boldsymbol{\lambda}) = \frac{\exp\left(\beta \sum_{l \inf} \lambda_l\right)}{\sum_{f'\in\mathcal{F}} \exp\left(\beta \sum_{l\in f'} \lambda_l\right)} \quad \forall f\in\mathcal{F}. \tag{21}$$

We first study the impact of solving the subproblem $\mathbf{MWIS}$ approximately by (19).

1) Entropy Term as a Consequence of Log–Sum–Exp Approximation: It is unlikely that we are still solving the original problem $\mathbf{MP}$. After we approximate problem $\mathbf{MWIS}$ by the problem in (20), the partial Lagrangian problem in (17) turns into

$$\begin{aligned} \min_{\boldsymbol{\lambda}\geq 0} \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} & \sum_{s\in S} U_s(z_s) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \\ & - \sum_{l\in L} \lambda_l \left(\sum_{s:l\in s, s\in R} z_s - \sum_{f:l\in f} p_f \right) \\ \text{s.t.} & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned} \tag{22}$$

One can verify that it is the partial Lagrangian problem of the following primal problem:

$$\begin{aligned} \mathbf{MP}-\mathbf{MA}: \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} & \sum_{s\in S} U_s(z_s) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \\ \text{s.t.} & \sum_{f:l\in f} p_f \geq \sum_{s:l\in s, s\in R} z_s \quad \forall l\in L, \\ & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned} \tag{23}$$

Comparing problems $\mathbf{MP}-\mathbf{MA}$ and $\mathbf{MP}$, we observe that when we approximate the subproblem $\mathbf{MWIS}$ by the one in (20), we are in effect approximating the problem $\mathbf{MP}$ by problem $\mathbf{MP}-\mathbf{MA}$, which has an additional entropy term in its objective function. We remark that the entropy term appears as a direct consequence of our approximating the max function with the log–sum–exp function in (19), independent of any wireless protocol, e.g., CSMA, to be used.

Historically, by modeling and studying the carrier sensing behavior, the authors of [3] showed that the percentage of the active time of independent sets, under the CSMA scheduling with transmission aggressive vector $\boldsymbol{\lambda}$, is given by $p_f(\boldsymbol{\lambda})$ in (21). The authors of [5], [6] then use reverse-engineering to show that $p_f(\boldsymbol{\lambda})$ in (21) is the optimal solution to the problem in (20). With this observation, the authors of [5], [6] design distributed algorithms on top of CSMA to solve problem $\mathbf{MP}-\mathbf{MA}$, with an optimality gap upper-bounded by $\frac{1}{\beta}\log\|\mathcal{F}\|$.

2) CSMA as Distributed Implementation of Markov Chain: From a forward engineering perspective, imagine that the CSMA protocol was not invented and did not exist yet. Then following the Markov approximation technique, we could have designed a time-reversible Markov chain whose stationary distribution is given by (21) and worked out its distributed implementation.

In particular, the states of the Markov chain are the independent sets over G_c. To make sure the network operates over only the independent sets, we could have worked out the implication that two interfering links (in particular their transmitters) must be able to sense each other so one would keep silent while the other is transmitting. This could be done distributedly by each transmitter sensing its receiving power and only starting its transmission if the power is below a properly selected threshold [11].

We now follow OPT4 (discussed in Section II-C) to design the transition rates. We start by only allowing direct transitions between two "adjacent" states (independent sets) f and f' that differ by one and only link. That is,

1) we set $q_{f,f'}$ to zero, unless $|f|-|f'| = \pm 1$. Here, $|\cdot|$ represents the size of a set.
   By this design, the transition from f to $f' = f\cup\{l'\}$ corresponds to link l' starting its transmission. Similarly, the transition from f' to f corresponds to link l' finishing its on-going transmission.
   Now, consider two states f and f' where $f' = f\cup\{l'\}$,
2) we set $q_{f',f}$ to 1, and
   $$q_{f,f'} = \exp\left(\beta\left(\sum_{l\in f'}\lambda_l - \sum_{l\in f}\lambda_l\right)\right) = \exp\left(\beta\lambda_{l'}\right).$$
   To achieve transition rate $q_{f,f'}$, the transmitter of link l' waits for a back-off time that follows exponential distribution with rate $\exp(\beta\lambda_{l'})$ before it starts to transmit. During the countdown, if the link l' (in particular its transmitter) senses another interfering link is in transmission, it will freeze its countdown process. When the transmission is over, link l' counts down according to the residual back-off time, which is still exponentially distributed with the same rate $\exp(\beta\lambda_{l'})$ because of the memoryless property of exponential distributions.

The transition rate only depends on the Lagrange multiplier $\lambda_{l'}$ (called transmission aggressiveness in [5]) and is proportional to the local queue length of link l', as discussed in [5] and [6] and in Section III-C.4.

Similarly, the transition rate $q_{f',f}$ can be achieved by link l' setting its transmission time to follow exponential distribution with unit rate.

In the end, this distributed implementation leads to the discovery of the CSMA protocol, with adjustable transmission aggressiveness. In fact, the Markov approximation technique is a general framework. Since Markov approximation generalized [5], [6], the enhanced CSMA protocol becomes a special application of Markov approximation. Also, using similar techniques, Markov approximation can lead to new algorithms for other network optimization problems, as illustrated in Sections IV and V.

3) Approximation Accuracy Limited by Physical Constraints: Mathematically, as β approaches infinity, we should be able to solve $\mathbf{MWIS}$ exactly. However, there are certain physical constraints preventing β from being too large. In CSMA networks, the value of $\exp(\beta\lambda_{l'})$ corresponds to the ratio of average packet-transmission duration to average backoff time [3]. For a given fixed packet-transmission duration (e.g., that corresponds to the maximum size of a wireless packet), increasing β basically means decreasing the average back off time. However, the average backoff time cannot be arbitrarily decreased. In practice, the back-off process is actually time-slotted. The slot length σ is constrained by circuit design considerations, or more fundamentally the radio propagation delay. For a WLAN in which the largest distance between two stations is d, as a rule of thumb $\sigma \geq 2d/c$ (the round-trip propagation delay) for CSMA to operate properly. If we assume the radius of a WLAN coverage is 75 m and the speed of light $c = 3 \times 10^8$ m/s, then $\sigma \geq 1$ μs. In 802.11b, $\sigma = 20$ μs; but here we will assume the fundamental limit of $\sigma = 1$ μs.

Typically, we should allow for a sufficient number of slots in the average backoff time to avoid excessive packet collisions due to simultaneous backoff countdown to zero by two or more transmitters. In 802.11b, for example, the average number of backoff countdown is around 15 time slots. Using this number, the average backoff time is therefore 15×1 μs $= 15$ μs.

Now suppose that the data rate of the WLAN is 10 Mbps, and the average packet size is 1 kB. The packet duration is then in the ballpark of 800 μs (ignoring DIFS and ACK). The largest possible value for $\exp(\beta\lambda_{l'})$ is then 800/15=53 (or $\beta\lambda_{l'} \leq 4$). In 802.11e, we could use the TXOP option to bundle packet transmissions together. Assume we bundle ten packets together for transmissions, then instead of 53, $\exp(\beta\lambda_{l'}) \leq 530$ (or $\beta\lambda_{l'} \leq 6.3$). That is, the backoff rate of link l' ($l' \in L$) is $\exp(\beta\lambda_{l'})$, which cannot go beyond 530.

4) Solving Problem $\mathbf{MP-MA}$ *by CSMA and Primal-dual Algorithm:* With the approximated optimal value to problem $\mathbf{MWIS}$ in (19), we can solve the following problem to obtain the optimal solution $\boldsymbol{z}^*$ and $\boldsymbol{\lambda}^*$ (and thus $\boldsymbol{p}^*$):

$$\min_{\boldsymbol{\lambda}\geq 0}\max_{\boldsymbol{z}\geq 0}\sum_{s\in S}U_s(z_s)-\sum_{l\in L}\lambda_l\sum_{s:l\in s,s\in S}z_s$$
$$+\frac{1}{\beta}\log\left[\sum_{f\in\mathcal{F}}\exp\left(\beta\sum_{l\in f}\lambda_l\right)\right]. \quad (24)$$

This problem can be solved by either a dual algorithm or a primal-dual algorithm. Dual algorithms has been studied for a slightly different formulation in [5] and [6]. We study a primal-dual algorithm as follows:

$$\begin{cases}\dot{\lambda}_l = k_l\left[\sum_{s:l\in s,s\in S}z_s-\sum_{l\in f}p_f(\beta\boldsymbol{\lambda})\right]^+_{\lambda_l}, \\ \dot{z}_s=\alpha_s\left[U_s'(z_s)-\sum_{l\in s}\lambda_l\right]^+_{z_s}\end{cases} \quad (25)$$

where $k_l (l \in L)$ and $\alpha_s (s \in S)$ are positive constants and function $[b]^+_a = \max(0,b)$ if $a \leq 0$ and equals b otherwise. The advantage of the primal-dual algorithm is that the changes in sending rate z_s (and correspondingly λ_l) is smoother than that in the dual algorithm.

Note that in (25), we do not need accurate estimates of the stationary distributions $p_f(\beta\boldsymbol{\lambda})$. In fact, $\sum_{l\in f}p_f(\beta\boldsymbol{\lambda})$ is the stationary throughput of link l and can be measured directly by running CSMA protocol network-wide with transmission aggressive vector $\beta\boldsymbol{\lambda}$. This is a key observation made in [3], [5], and [6]. The Lagrange dual variable λ_l can then be updated based on information of the local queue at link l.

The proof of the convergence of the algorithm in (25) can be based on a standard technique using Lyapunov function [12], assuming the distribution of Markov chain states converges to the stationary distribution instantaneously.

In practice, however, the distribution of Markov chain states may not converge before the primal-dual algorithm (25) evolves. The algorithm then turns into a stochastic primal-dual algorithm, given as follows: for all $l \in L$ and $s \in S$, let

$$\begin{cases}\lambda_l(m+1)=\left[\lambda_l(m)+\epsilon(m)\left(\sum_{s:l\in s,s\in S}z_s(m)-\bar{\theta}_l(m)\right)\right]^+, \\ z_s(m+1)=\left[z_s(m)+\epsilon(m)\left(U_s'(z_s(m))-\sum_{l:l\in s}\lambda_l(m)\right)\right]^+,\end{cases} \quad (26)$$

where $[\cdot]_+ \triangleq \max(\cdot,0)$, $\epsilon(m)$ is the step size, and $\bar{\theta}_l(m)$ is the average link rate measured by link l within the update interval T_m. When $\epsilon(m)$ is very small, the primal-dual algorithm (25) is a continuous time approximation of (26).

Under suitable choices of step sizes and update intervals, we establish the convergence of the stochastic primal-dual algorithm (26) with probability 1 in the following theorem.

Theorem 2: Assume that $U_s'(0) < \infty$ $\forall s \in S$, $\max_{s,m} z_s(m) < \infty$ and $\max_{l,m}\lambda_l(m) < \infty$. Then, the stochastic primal-dual algorithm in (26) converges to the optimal solutions of $\mathbf{MP-MA}$ asymptotically with probability 1 under the following conditions on step sizes and update intervals:

$$T_{m+1} \geq T_m \ \forall m \geq 1, \quad (27a)$$
$$\epsilon(m) > 0 \ \forall m \geq 1, \quad (27b)$$
$$\sum_{m=1}^{\infty}\epsilon(m)=\infty, \quad (27c)$$
$$\sum_{m=1}^{\infty}\epsilon^2(m)<\infty, \quad (27d)$$
$$\sum_{m=1}^{\infty}\frac{\epsilon(m)}{T_m}<\infty. \quad (27e)$$

Further, the setting $\epsilon(1) = T(1) = 1$, $\epsilon(m) = \frac{1}{m}$, $T_m = m$, $m \geq 2$ is one specific choice of step sizes and update intervals satisfying conditions (27a)–(27e). This setting depends only on the index m, and thus can be to any networks.

The proof is relegated to Appendix-B. Inspired by and similar to [5], [13], [14], we also adopt the standard methods of stochastic approximation [15] and Markov chain [16], [17] in establishing our convergence results.

D. Discussions

The differences between our work and [5], [13], [14] are the following.

First, we adopt a link-centric formulation in approaching the problem and designing the algorithms, while [5], [13], [14] adopt a node-centric formulation. Theoretically, there are pros and cons of adopting these two formulations, as discussed in [1]. We adopt the link-centric formulation because of the following practical concern. In both formulations, the Lagrange multipliers (prices) are interpreted as a scalar multiple of the queue length. In the link-centric formulation, each link only needs to maintain one queue. In contrast, in the node-centric formulation, each node needs to maintain multiple queues, one per flow. Consequently, solutions following the node-centric formulation require each node to set up and maintain per-flow state information. This incurs substantially more overheads than the solutions following the link-centric formulation, especially in large wireless networks with many flows dynamically coming and going.

Second, we adopt a primal-dual algorithm while [5], [13], [14] adopt dual algorithms. Given our link-centric formulation, for dual algorithms, the source rate is a function of link prices and is sensitive to the measurement noises of the link prices [18, Ch. 3]. In contrast, for primal-dual algorithms, the source rate is adjusted gradually. A sudden change in the measured price (due to noises) will not result in an abrupt change in the source rate. Consequently, the source rate controlled by primal-dual algorithms is smoother than that controlled by dual algorithms, especially in the presence of measurement noises.

Third, our proof studies the saddle points of Lagrangian function, while [5], [13], [14] study the optimal dual solutions directly. As a result, our stability proofs involve the design and use of Lyapunov functions different from those used in [5], [13], and [14].

IV. Case 2: Path Selection in Wireline Networks

A. Settings

Consider a wireline network $G = (V, L)$, where V and L are the sets of nodes and links, respectively. The capacity of link $l \in L$ is denoted by C_l. Let J_s denote the set of paths available for user $s \in S$. For each path that a user s selects from J_s, it opens a connection to transfer data. Maintaining connections and paths consume users' resources and incur overheads. Due to the limited system resource or the overhead concern, each user $s \in S$ operates over at most D_s paths, where D_s is a nonnegative integer.

Let $\mathcal{F}$ denote the set of all possible configurations of paths used by users. A configuration $f \in \mathcal{F}$ represents the set of paths used by all $s \in S$. Given an $f \in \mathcal{F}$, we denote those used by user s by $J_{s,f} \subseteq J_s$, where $|J_{s,f}| = D_s$. We assume the utility functions to be twice differentiable, increasing and strictly concave.

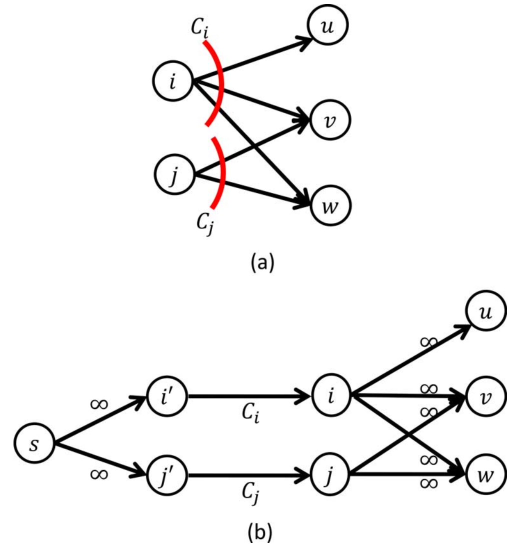

Fig. 2. (a) Small-scale P2P file sharing scenario, where three peers u, v, and w want to download a file from two peers i and j. Peers i and j have the file and have the uplink capacities of C_i and C_j, respectively. (b) Path-based equivalent representation for the P2P file sharing scenario in (a). Peers with upload capacity constraints (b) path-based equivalent representation.

Similar to [19], we also make a *single-bottleneck assumption*. That is, we assume that for each path in $\mathcal{F}$, there exists at most one bottleneck along this path. Here "bottleneck" is defined as the finite-capacity link shared among multiple paths. Therefore, at most one finite-capacity link of a path will be shared with other paths. While this might seem to be a limited assumption, it is a reasonable model for commonly studied peer-to-peer (P2P) scenarios, as we explain in next section.

B. Justification of the Single-Bottleneck Assumption

We show that this assumption is a reasonable model for commonly studied P2P scenarios. In P2P system analysis and design, it is common to assume that peer uplinks are the capacity bottleneck in P2P networks [20], [21], [22]. This assumption can be partially justified by the empirical observation that as residential broadband connections with asymmetric upload and download rates become increasingly dominant, bottlenecks typically are at the uplinks of the access networks rather than in the middle of the Internet.

In such P2P scenarios, an important resource allocation problem is to allocate peers' uplink capacities to serve multiple neighboring peers. This uplink capacity-allocation problem can be formulated as a multipath resource allocation problem under the single-bottleneck assumption. An illustrating example is shown in Fig. 2.

Fig. 2(a) shows a small-scale P2P file sharing scenario. In the figure, there are five peers; the edges represent TCP/UDP connections that the peers use to communicate. Three peers u, v, and w want to download a file from two peers i and j. Two peers i and j have the file and have upload capacities of C_i and C_j, respectively. That is, the total amount of data that peer i

(and j) sends to peers u, v, and w in a unit time cannot exceed C_i (and C_j).

Fig. 2(b) shows an equivalent representation of the same scenario as in Fig. 2(a). In the figure, we "split" peers i and j into two pair of nodes i and i', and j and j', respectively. We also add a "virtual" source node s and several edges; the topology and the corresponding link capacities are shown in the figure. The upload capacities of peer i and j are modeled by the edges (i', i) with capacity C_i and (j', j) with capacity C_j, respectively.

In the new representation, there are five paths in total from the virtual source s to peers u, v, and w. Peers u, v, and w want to receive the same file from s using the multiple paths between s and itself. For instance, peer v wants to receive the file from s by using two paths $s \to i' \to i \to v$ and $s \to j' \to j \to v$. Given this representation, allocating the upload capacities of peers i and j is equivalent to allocating the capacities of edges (i', i) and (j', j), respectively.

By comparing the scenarios in Fig. 2(a) and (b), we can see the problem of allocating the upload capacities of peers i and j to serve peers u, v, and w in Fig. 2(a) can be formulated as a multipath resource allocation problem under the single-bottleneck assumption in Fig. 2(b). Therefore, the single-bottleneck assumption is a reasonable model for allocating upload capacities in P2P file sharing scenarios.

C. Joint Path Selection and Multipath Utility Maximization

Now we consider the following utility maximization problem based on path selection, where we time-share among a set of configurations to maximize the aggregate user utility

$$\begin{aligned}\mathbf{PS}: \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} & \sum_{s\in S} U_s(z_s) \\ \text{s.t. } z_s &\leq \sum_{f\in\mathcal{F}} R_{s,f}\, p_f \quad \forall s\in S, \\ & \sum_{f\in\mathcal{F}} p_f = 1, \end{aligned} \tag{28}$$

where z_s is the long-term throughput of user $s \in S$, p_f is the probability (or time fraction) of the configuration f, and $R_{s,f}$ is called the "equilibrium rate" for user s in configuration f. It is the aggregate rate source s obtained at the optimal solution to the following multipath utility maximization problem with uncoordinated congestion control [19]:

$$\begin{aligned}\mathbf{MP}-\mathbf{UCC}: \quad \max\nolimits_{\boldsymbol{y}\geq\mathbf{0}} \quad & \sum_{s\in S}\sum_{j\in J_{s,f}} U_s(y_j) \\ \text{s.t.} \quad & \sum_{j:l\in j} y_j \leq C_l \quad \forall l \in L_f, \end{aligned} \tag{29}$$

where L_f is the set of links used by all users under configuration f, y_j is the path rate for path $j \in J_{s,f}$, $s \in S$, and $\boldsymbol{y} = [y_j\ \forall j \in J_{s,f}, s \in S]^T$ is the vector of rates of all paths. Let optimal solutions of the problem $\mathbf{MP}-\mathbf{UCC}$ denoted by $\hat{y}_j$, $j \in J_{s,f}$, $s \in S$, the equilibrium capacity is given by

$$R_{s,f} = \sum_{j\in J_{s,f}} \hat{y}_j. \tag{30}$$

By (30), we implicitly assume a timescale separation between solving the problems $\mathbf{MP}-\mathbf{UCC}$ and $\mathbf{PS}$ [19]. This assumption is justified to some extent by the following observations. Given the configuration $f \in \mathcal{F}$, problem $\mathbf{MP}-\mathbf{UCC}$ can be solved by standard distributed flow control algorithms [19], in a timescale on the order of round trip time. On the other hand, the path selection is likely to operate at a much slower timescale due to the overheads involved in configuring paths and setting up connections.

With the two timescale separation in place, we focus on solving the combinatorial problem $\mathbf{PS}$ in the slow timescale.

D. Markov Approximation

Following a procedure similar to that in Section III, we apply Markov approximation and at the end turn to solve an approximated version of problem $\mathbf{PS}$ as follows:

$$\begin{aligned}\mathbf{PS}-\mathbf{MA}: \quad \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \quad & \sum_{s\in S} U_s(z_s) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \\ \text{s.t.} \quad & z_s \leq \sum_{f\in\mathcal{F}} R_{s,f}\, p_f \quad \forall s \in S, \\ & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned} \tag{31}$$

To proceed, we relax the first set of constraints and denote $\boldsymbol{\lambda} = [\lambda_s, s \in S]$ as the vector of Lagrange multipliers. Following an analysis similar to that in Section III, the optimal solution of the problem in (31) can be obtained by searching the saddle point of the following function:

$$\sum_{s\in S}[U_s(z_s) - \lambda_s z_s] + \frac{1}{\beta}\log\left[\sum_{f\in\mathcal{F}}\exp\left(\beta\sum_{s\in S} R_{s,f}\,\lambda_s\right)\right] \tag{32}$$

and at the same time setting

$$p_f(\beta\boldsymbol{\lambda}) = \frac{\exp\left(\beta\sum_{s\in S} R_{s,f}\,\lambda_s\right)}{\sum_{f\in\mathcal{F}}\exp\left(\beta\sum_{s\in S} R_{s,f}\,\lambda_s\right)}\ \forall f \in \mathcal{F}. \tag{33}$$

We explore algorithm design based on this observation in Sections IV-E and IV-F. The $p_f(\beta\boldsymbol{\lambda})$ in (33) can be interpreted as the stationary distribution of a time reversible Markov chain, whose states are the configurations in $\mathcal{F}$. We first discuss how to design and implement such a Markov chain in a distributed manner, and then design stochastic algorithms to pursue the saddle point of the function in (32).

E. Design and Implementation of Markov Chain

Recall that a configuration $f \in \mathcal{F}$ represents the set of paths used by all users $s \in S$. First, we set the transition rate $q_{f,f'}$ between two configurations f and f' to be zero, unless f and f' satisfy that

C1: $|f \cup f' - f \cap f'| = 2$;

C2: there exists a user, denoted by $s(f, f')$, such that $f \cup f' - f \cap f' \in J_{s(f,f')}$.

This way, the transition from f to f' corresponds to a single user $s(f, f')$ switching a single path.

Second, for f and f' that satisfy **C1** and **C2**, we follow OPT1 discussed in Section II-C to design their transition rate $q_{f,f'}$. Direct implementation of OPT1, however, usually requires user $s(f,f')$ to know global information $\sum_{s\in S} R_{s,f}\lambda_s$, a term difficult to acquire in practice. To this extent, we find that a unique structure of our problem can simplify the implementation.

We introduce a new concept to facilitate our discussions. Given a path j, its *neighboring path set* $\mathcal{N}(j)$ is defined as the set of paths that share links with j, i.e., $\mathcal{N}(j) = \{j' : j' \cap j \neq \emptyset\}$. Note that here we also include path j itself in $\mathcal{N}(j)$ for convenience. Since there is at most one bottleneck link per path, we have (i) only one link of path j is shared with other paths in $\mathcal{N}(j)$; (ii) this particular link must be the only bottleneck link of any path $j' \in \mathcal{N}(j)$. Consequently, all paths in $\mathcal{N}(j)$ have identical neighboring set, i.e., $\mathcal{N}(j') = \mathcal{N}(j)$ for all j' in $N(j)$. For any path $j' \notin \mathcal{N}(j)$, $\mathcal{N}(j') \cap \mathcal{N}(j) = \emptyset$.

For example, in Fig. 2(b), for a path $s \to i' \to i \to v$, its neighbor path set is given by,

$$\begin{aligned}\mathcal{N}(s \to i' \to i \to v) = \{&s \to i' \to i \to v,\\ &s \to i' \to i \to u,\\ &s \to i' \to i \to w\}.\end{aligned}$$

Only one link (i', i) of path $s \to i' \to i \to v$ is shared with other paths in $\mathcal{N}(s \to i' \to i \to v)$. This link (i', i) is also the only bottleneck link of any path in $\mathcal{N}(s \to i' \to i \to v)$. It can also be verified that all the paths in $\mathcal{N}(s \to i' \to i \to v)$ have the identical neighboring set.

Then we have the following observation.

Lemma 2: Under the setting of uncoordinated congestion control,

$$R_{s',f} = R_{s',f'}, \text{ if } J_{s',f} = J_{s',f'} \text{ and } J_{s',f} \cap \mathcal{N}(j') = \emptyset \\ \forall j' \in (f \cup f' - f \cap f').$$

Proof: Under the setting of uncoordinated congestion control, each path has its own utility function to maximize. Two paths are independent to each other if they are disjoint. Therefore, the optimal rate of path j depends only on its neighboring paths set $\mathcal{N}(j)$. If the user s' does not change paths, and all its paths are disjoint with paths in the set $\bigcup_{j'} \mathcal{N}(j'), j' \in f \cup f' - f \cap f'$, then paths in $J_{s',f}$ and their neighboring path sets will not be affected by path swapping. Thus by (29) and (30), the equilibrium rate $R_{s',f}$ is invariant and equals to $R_{s',f'}$. Therefore, the equilibrium rates of a user s' under f and f' are the same if s' does not change paths, and for any path $j' \in f \cup f' - f \cap f'$, all paths of s' do not belong to the neighboring path set of j'. ■

Let $H(f,f')$ be the set of such "invariant" users under configurations f and f', i.e., $H(f,f') = \{s' : J_{s',f} = J_{s',f'}$ and $J_{s',f} \cap \mathcal{N}(j') = \emptyset\ \forall j' \in (f \cup f' - f \cap f')\}$. Then, to satisfy the detailed balance equation $q_{f,f'} p_f(\beta\boldsymbol{\lambda}) = q_{f',f} p_{f'}(\beta\boldsymbol{\lambda})$ for f and f' that satisfy **C1** and **C2**, it is sufficient to let

$$\begin{cases} q_{f,f'} = \left[\exp\left(\beta \sum_{s\in S-H(f,f')} R_{s,f}\lambda_s\right)\right]^{-1}, \\ q_{f',f} = \left[\exp\left(\beta \sum_{s\in S-H(f,f')} R_{s,f'}\lambda_s\right)\right]^{-1}. \end{cases} \tag{34}$$

The common part $\exp\left(\beta \sum_{s\in H(f,f')} R_{s,f}\lambda_s\right)$ appears on both sides of the detailed balance equation and gets canceled. Now, to implement transition rate $q_{f,f'}$ in (34), the user $s(f,f)$ needs to collect the information $R_{s,f}\lambda_s$ from s in $S - H(f,f')$.

Note that $S - H(f,f')$ is the set of users whose paths share links with $s(f,f')$. Users s in $S - H(f,f')$ can then leave the information $R_{s,f}\lambda_s$ at each router, and user $s(f,f')$ can collect the $R_{s,f}\lambda_s$ left by all users s currently using the path via a feedback mechanism (e.g., through the ACK for the packets of $s(f,f')$). The shared routers can be regarded as the shared memory between $s(f,f')$ and s in $S - H(f,f')$. In this way, $s(f,f')$ acquires the needed information to compute $q_{f,f'}$ and $q_{f',f}$ in (34) in a distributed manner.

We briefly describe the distributed implementation as follows.

Stage 0. Initially, every user s randomly selects D_s paths from its path set J_s.

Stage 1. Denote the current configuration as f. Each user s computes

$$\begin{aligned} q_s &= \sum_{f':s(f,f')=s} q_{f,f'} \\ &= \sum_{f':s(f,f')=s} [\exp(\sum_{s'\in S-H(f,f')} \beta R_{s',f}\lambda_{s'})]^{-1}. \end{aligned} \tag{35}$$

User s then counts down according to an exponential distribution with parameter q_s. Here for each f' satisfying $s(f,f') = s$, the information $\sum_{s'\in S-H(f,f')} R_{s',f}\lambda_{s'}$ can be acquired in the following way. Note that each user s needs to continuously receive the feedback information on all paths in J_s, not just the paths that it is currently using. Therefore, "probing packets" must also be sent along the paths that are not currently being used so that ACK can be triggered to collect the aggregated information in the routers along the path. Thus, for each user s' in S and each active path $j \in J_{s',f}$ (path that is currently used by s'), user s' adds a header containing $R_{s',f}\lambda_{s'}$ to data packets before sending them out along path j. On the other hand, for each user s' in S and each inactive path $j \in J_s - J_{s',f}$ (path that is not currently used by s'), user s' needs to send probing packets without putting $R_{s',f}\lambda'_s$ in the headers on path j occasionally (i.e., not as frequently as ordinary data packets). The only purpose of probing packets is to trigger ACK in the reverse direction. Then every router on path $j \in J_{s'}$ records the information of $R_{s',f}\lambda_{s'}$ for every s' whose traffic (either data packets or probing packets) passing through them. Assuming the traffic in the reverse direction (e.g., ACK packets for either data packets or probing packets) uses the same paths as the forward direction traffic, the ACK packets can collect the $R_{s',f}\lambda_{s'}$ $(s' \in S - H(f,f'))$ information from the routers on their way to user s.

Stage 2. During the countdown, each user s also continuously monitors whether other users sharing links with it undertake a path swapping. This can be done by the users who swap paths leave a 1 bit of "swapped" information at the routers, and all users whose traffic passing by this router can collect this bit of information. If a user s detects a path

swapping, it will reset its counter and jump to *Stage 1*. Each router resets the "swapped" bit once all users passing by it have collected the information using their ACK packages.
Stage 3. When user s's countdown expires, it selects the target configuration $f'(s(f,f') = s)$ with probability $q_{f,f'}/q_s$ and swaps the corresponding two paths. The whole system transits to *Stage 1*.

Remarks: (i) We observe nontrivial overheads incurred during the implementation. However, it is a mild burden for some practical systems. For example, in the motivating example of P2P networks (shown in Fig. 2), our implementation in fact requires each user (peer) to continuously receive feedback information from all its neighboring nodes in J_s. This is what current P2P systems (e.g., BitTorrent systems [23]) already do in practice. (ii) By a process very similar to that in Section III-C.4, we can prove that the above implementation realizes a time-reversible Markov chain with stationary distribution in (33); hence, we omit the details.

F. *Solving Problem* $\mathbf{PS-MA}$ *With a Primal-Dual Algorithm Over a Markov Chain*

Following a procedure similar to that in Section III-C.4, we design a distributed stochastic primal-dual algorithm to pursue the saddle points of the function in (32), on top of the Markov chain implemented in the previous section. Specifically, for all $s \in S$, we let

$$z_s(m+1) = \left[z_s(m) + \epsilon(m)(U'_s(z_s(m)) - \lambda_s(m))\right]^+, \tag{36}$$

$$\lambda_s(m+1) = \left[\lambda_s(m) - \epsilon(m)\left(\bar{\theta}_s(m) - z_s(m)\right)\right]^+, \tag{37}$$

where $[\cdot]^+ \triangleq \max(\cdot, 0)$, $\epsilon(m)$ is the step size, and $\bar{\theta}_s(m)$ is the average service rate user s actually obtains within the update interval T_m.

Under suitable choices of step sizes and update intervals, we establish the convergence of the stochastic primal-dual algorithm (36)–(37) with probability 1 in the following theorem.

Theorem 3: Assume that $U'_s(0) < \infty \ \forall s \in S$, $\max_{s,m} z_s(m) < \infty$ and $\max_{s,m} \lambda_s(m) < \infty$. Then, the stochastic primal-dual algorithm in (36)–(37) converges to the optimal solution of problem $\mathbf{PS-MA}$ with probability 1 if the following conditions for step sizes and update intervals hold:

$$T_{m+1} \geq T_m \ \forall m \geq 1, \tag{38a}$$

$$\epsilon(m) > 0 \ \forall m \geq 1, \tag{38b}$$

$$\sum_{m=1}^{\infty} \epsilon(m) = \infty, \tag{38c}$$

$$\sum_{m=1}^{\infty} \epsilon^2(m) < \infty, \tag{38d}$$

$$\sum_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty. \tag{38e}$$

Further, the setting $\epsilon(1) = T(1) = 1$, $\epsilon(m) = \frac{1}{m}$, $T_m = m$, $m \geq 2$ is one specific choice of step sizes and update intervals satisfying conditions (38a)–(38e).

We omit the details of the proof of Theorem 3 since it is similar to that of Theorem 2.

V. CASE 3: CHANNEL ASSIGNMENT IN WIRELESS LANS

A. *Settings*

Consider a wireless LAN with N access points (AP). Each AP is associated with a set of clients that access the Internet via this AP. In our setting, APs are connected by a wireline backbone, e.g., Ethernet, so that they can communicate with each other with negligible cost. This corresponds to the case where APs belong to the same administrative zone and can coordinate among themselves. Each AP can choose one channel to operate from a set of M available channels, denoted by $C = \{c_1, c_2, \ldots c_M\}$. We define a channel-assignment configuration as the vector indicating the channel choice of every APs, i.e., $f \triangleq [f_1, f_2, \ldots, f_N]$, where $f_i \in C$ denotes the channel choice of the ith AP. Let $\mathcal{F}$ be the set of all feasible f.

In our setting, given a configuration f, the wireless stations compete to access the wireless channels according to standard 802.11 protocol. In particular, what we consider is the case where the MAC-layer link scheduling, subject to wireless interference constraints, is done by certain practical MAC-layer protocol. We focus on network-layer frequency-channel assignment to optimize the overall wireless-network performance. We consider the problem we study in this section a network-layer design problem since it concerns about resource allocation at a timescale much slower than that of MAC-layer scheduling. In contrast, the problem studied in Section III (i.e., Case 1) concerns MAC-layer design.

We denote the downlink throughput observed by AP i under configuration f by R_i^f. Upon observing R_i^f, AP i obtains a utility of $U_i\left(R_i^f\right)$. We assume function U_i to be strictly increasing and concave, and twice differentiable. Further, without loss of generality, we assume

$$U_i(R_i^f) \in [U_{\min}, U_{\max}] \ \forall i \in \{1, \ldots, N\}, f \in \mathcal{F}, \tag{39}$$

where $U_{\min}$ and $U_{\max}$ are finite constants.

The problem of finding the best channel assignment to maximize normalized system-wide utility is as follows:

$$\mathbf{CA}: \quad \max_{f \in \mathcal{F}} \quad x_f, \tag{40}$$

where $x_f = \frac{1}{N}\sum_{i=1}^{N} U_i(R_i^f)$ is the normalized system utility under channel configuration f for any $f \in \mathcal{F}$. By (39) we know $U_{\min} \leq x_f \leq U_{\max} \ \forall f \in \mathcal{F}$.

Remarks: This problem is a combinatorial problem, and the size of feasible set $\mathcal{F}$ is very large even for a network of modest size, making the problem hard to solve. Furthermore, even if we could handle problems of this size, we may not know R_i^f *a priori* because they can only be measured in real time in the field, and accurate analytical estimates of them are lacking.[6] Therefore, we assume a measurement-based approach in which R_i^f is obtained from real-time measurements. We also assume that the measurement interval is much smaller than the timescale on which the APs perform channel assignments.

[6]Indeed, the problem of finding the analytical expression of R_i^f that fully accounts for the effects of the carrier-sensing relationships among the links, hidden-node effects, back-off collisions, and channel fading is particularly challenging and largely open.

Let p_f be the fraction of the time that configuration f is activated, i.e., AP i chooses channel f_i. We reformulate problem $\mathbf{CA}$ as follows:

$$\mathbf{CA-AVG}: \quad \max_{p\geq 0} \quad \sum_{f\in\mathcal{F}} p_f x_f \qquad \text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1. \tag{41}$$

We remark that the problem $\mathbf{CA-AVG}$ is still hard to solve as the number of variables is still combinatorial.

B. Markov Approximation

We apply Markov approximation and turn $\mathbf{CA-AVG}$ to the following $\mathbf{CA-MA}$ optimization problem:

$$\mathbf{CA-MA}: \quad \max_{p\geq 0} \quad \sum_{f\in\mathcal{F}} p_f x_f - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \qquad \text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1. \tag{42}$$

Its optimal solution is given by

$$p_f^* = \frac{\exp(\beta\cdot x_f)}{\sum_{f'\in\mathcal{F}}\exp(\beta\cdot x_{f'})}\ \forall f\in\mathcal{F}. \tag{43}$$

We consider a continuous time-reversible Markov chain that has the stationary distribution as p_f^* $(f\in\mathcal{F})$. We call it a channel-hopping Markov chain. Its states are the feasible configurations. Let $q_{f,f'}$ and $q_{f',f}$ be the transition rates from a state f to another state f'. To achieve the desired stationary distribution, we follow OPT1 discussed in Section II-C, and set

$$q_{f,f'} = \alpha\left[\exp(\beta\cdot x_f)\right]^{-1}. \tag{44}$$

We do not consider OPT2–4 because they all involve probing the performance of the target configuration before making the channel hopping decision, complicating the system design.

C. Implementation

We implement a channel-hopping Markov chain with transition rates in (44) as follows. Initially, the APs randomly pick their channels. Under current configuration f, each AP keeps track of its own $U_i(R_i^f)$ based on the measurement of R_i^f within a constant measurement interval T_m, and periodically broadcasts it to all other APs. This broadcast can be done using the backbone Ethernet connecting the APs.

Each AP also generates an exponentially distributed random number with mean equal to

$$\frac{\exp(\beta\cdot x_f)}{\alpha(M-1)} \tag{45}$$

and counts down according to this number. When the countdown of an AP expires, this AP *randomly* switches to one of its $(M-1)$ not-in-use channels. This AP also informs the other APs to terminate their current countdown processes and start fresh ones using new measurements under the new configuration f'. We name this implementation "Wait-and-Hop" algorithm for the ease of reference.

In the "Wait-and-Hop" algorithm, each AP runs a procedure which operates according to the state machine shown in Fig. 3.

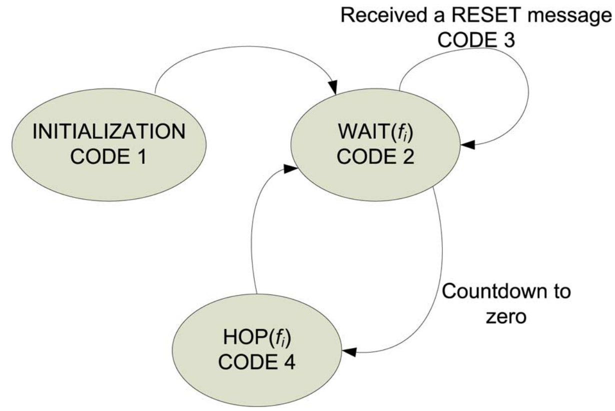

Fig. 3. State machine of a particular link in "Wait-and-Hop" algorithm.

We focus on a particular AP i. The pseudocode under each state is shown in Algorithm 1.

Algorithm 1 "Wait-and-Hop" algorithm

$C = \{c_1, c_2, \ldots c_M\}$

Procedure CODE 1

  $f_i \leftarrow$ a channel randomly picked from C

  Transit to State WAIT (f_i)

end procedure

Procedure CODE 2

  Within the measurement interval T_m, measure R_i^f, compute and periodically broadcast $U_i\left(R_i^f\right)$

  Collect $U_j\left(R_j^f\right), j\in\{\{1,\cdots,N\}-\{i\}\}$ and compute x_f

  Generate a timer with initial value τ_i that follows exponential distribution with mean equal to $\exp(\beta\cdot x_f)/(\alpha(M-1))$ and begin counting down

end procedure

procedure CODE 3

  Terminate its current countdown process

  Transit to State WAIT (f_i)

end procedure

procedure CODE 4

  $f_i \leftarrow$ a channel randomly picked from $C-\{f_i\}$

  Broadcast a RESET message to other APs

end procedure

The correctness of the "Wait-and-Hop" implementation is shown in the following proposition.

Proposition 2: The "Wait-and-Hop" algorithm shown in Algorithm 1 realizes a continuous-time channel-hopping Markov chain with stationary distribution shown in (43).

The proof is relegated to Appendix-C.

1) Performance Analysis: In this section, we study the impacts of two parameters on the system performance. One is T_m, the constant measurement interval. The other is α, the scale coefficient of transition rates for the channel-hopping Markov chain.

We first study the impact of T_m on the optimality gap of system utilities.

If the measurement interval T_m is not long enough for the underlying CSMA Markov chain to converge to its steady-state distribution, then our measured value of R_i^f under the current configuration f is not accurate, but with certain errors. Utilizing Markov chain mixing time analysis for the underlying CSMA Markov chain, we can characterize the relationship between such measurement errors and the measurement interval T_m.

With errors in measured value of R_i^f, the transition rates of the channel-hopping Markov chain are *perturbed* and are not the same as the designed ones. It is straightforward to verify that the perturbed process is still a Markov chain and it has a stationary distribution, denoted by $\bar{\boldsymbol{p}} = (\bar{p}_f, f \in \mathcal{F})$.

Since the perturbed Markov chain has a different set of transition rates as compared to the designed ones, its stationary distribution $\bar{\boldsymbol{p}}$ may not be the same as the desired one $\boldsymbol{p}^*$ shown in (43). We adopt the total variation distance metric [17] to characterize the difference between them. The total variation distance between $\boldsymbol{p}^*$ and $\bar{\boldsymbol{p}}$ is given by

$$\|\boldsymbol{p}^* - \bar{\boldsymbol{p}}\|_{TV} \triangleq \frac{1}{2}\sum_{f\in\mathcal{F}} |p_f^* - \bar{p}_f|. \tag{46}$$

By adopting the perturbation analysis proposed in [24], we have the following result.

Theorem 4: $\|\boldsymbol{p}^* - \bar{\boldsymbol{p}}\|_{TV}$ is bounded as follows:

$$0 \le \|\boldsymbol{p}^* - \bar{\boldsymbol{p}}\|_{TV} \le 1 - \exp\left(-\frac{2\beta\phi U_{\max}}{T_m}\right), \tag{47}$$

where

$$\phi = L \cdot \left[\frac{5}{2}L(4 + \log 2) + 4\right] \tag{48}$$

and L is the number of links in underlying CSMA networks. Further, the optimality gap of system utilities is bounded as below:

$$|\boldsymbol{p}^*\boldsymbol{x}^T - \bar{\boldsymbol{p}}\boldsymbol{x}^T| \le 2U_{\max}\left(1 - \exp\left(-\frac{2\beta\phi U_{\max}}{T_m}\right)\right). \tag{49}$$

The proof is relegated to Appendix-D.

We have the following observations.

1) The upper bound on optimality gap shown in (49) is quite general, as it is independent of the number of channel configurations $|\mathcal{F}|$, the distribution of measurement errors, and the number of quantization levels of measurement errors.
2) The upper bound on optimality gap shown in (49) increases as β and ϕ increase. By (44), we know that the ratio of the transition rate with measurement errors to the transition rate without measurement errors increases as β increases. Intuitively, the stationary distribution of perturbed Markov chain deviates more from the desired distribution in (43), resulting in a larger optimality gap. On the other hand, by (48) we know that ϕ is determined by L, the number of links in underlying CSMA network. For a large CSMA network with large L, usually the corresponding CSMA Markov chain converges slowly. Given the measurement interval, the increasing convergence time of the CSMA Markov chain induces a larger measurement error and a larger optimality gap.
3) The upper bound on optimality gap shown in (49) decreases as T_m increases. Intuitively, the larger the value of T_m, the more accurate the measured rates and the less the measurement errors. Thus, optimality gap decreases. While increasing the value of T_m reduces the optimality gap, it also increases the measurement interval. As a result, it takes longer for the overall process to converge to the stationary performance. Thus by changing the values of T_m, we can trade between optimality gap and convergence time of overall process. We conduct numerical experiments to verify this tradeoff in Section V-E.4.

In the rest of this section, we study the impact of α on system overhead and convergence time of the overall channel hopping process. We know that the time interval between every two consecutive transitions of channel hopping process includes two parts: a constant measurement interval T_m and an exponentially distributed countdown time depending on α. Thus, convergence time of the overall channel hopping process depends on values of both T_m and α. However, given nonzero values of T_m, the overall channel hopping process is not Markovian. It is well known that it is open to obtain nontrivial analytical results on convergence time of non-Markovian processes [16], [25].

To proceed with the analysis and obtain insights on how α affects the performance, we assume $T_m = 0$ and there are no measurement errors.

Later by simulation results in Section V-E.5, we will show that scenarios with $T_m \neq 0$ have similar behaviors to the scenario with $T_m = 0$ on the tradeoff between system overhead and convergence time of channel-hopping Markov chain. As these simulation results suggest, it will be interesting to confirm these similar behaviors by conducting the theoretical analysis of scenarios with $T_m \neq 0$ in the future.

On one hand, we study the impact of α on system overhead. Without loss of generality, we assume the convergence time of the channel-hopping Markov chain is denoted by T_v, the number of channel configuration transitions is denoted by Θ, and the sequence of channel configuration transitions is denoted by $f_1, f_2, \ldots, f_\Theta$. Let T_i denote the time that Markov chain stays for the ith transition, for any $i \in \{1, 2, \ldots, \Theta\}$. Since $T_m = 0$, we have

$$T_v = \sum_{i=1}^{\Theta} T_i. \tag{50}$$

By (45), we know

$$E(T_i) = \frac{\exp(\beta x_{f_i})}{\alpha(M-1)} \; \forall i \in \{1, 2, \ldots, \Theta\}. \tag{51}$$

Therefore, we have

$$E(T_v) = \frac{\sum_{i=1}^{\Theta}\exp(\beta x_{f_i})}{\alpha(M-1)}. \tag{52}$$

We know the system overhead (e.g., the number of broadcasting messages) is proportional to the number of channel configuration transitions Θ. We are concerned with broadcast message intensity, defined as follows:

$$\rho = \frac{\Theta}{E(T_v)} = \frac{\alpha(M-1)}{\frac{1}{\Theta}\sum_{i=1}^{\Theta}\exp(\beta x_{f_i})}. \quad (53)$$

It is not hard to see that ρ increases as α increases. Intuitively, as α increases, the expected countdown time of an AP decreases, leading to the increasing frequency of broadcasting messages to all other APs. Therefore, the system overhead (the number of broadcasting messages per unit time) increases. Numerical experiments have been conducted to verify this property in Section V-E.5).

On the other hand, we study the impact of α on convergence time of the channel-hopping Markov chain. Let $\boldsymbol{H}_t(f)$ denote the probability distribution of all states in $\mathcal{F}$ at time t given that the initial state is f. Then, the mixing time (convergence time) of the channel-hopping Markov chain is defined as follows [25]:

$$t_{\text{mix}}(\epsilon) \triangleq \inf\left\{t \geq 0 : \max_{f\in\mathcal{F}} \|\boldsymbol{H}_t(f) - \boldsymbol{p}^{*}\|_{TV} \leq \epsilon\right\}. \quad (54)$$

To illustrate the relationship between $t_{\text{mix}}(\epsilon)$ and α, we adopt the spectral analysis method [17] to obtain a lower bound of $t_{\text{mix}}(\epsilon)$ and path coupling method [26] to obtain an upper bound of $t_{\text{mix}}(\epsilon)$. Corresponding results are shown as follows.

Theorem 5: The mixing time (convergence time) of the channel-hopping Markov chain is bounded as follows:

a) for general $\beta \in (0, \infty)$

$$t_{\text{mix}}(\epsilon) \geq \frac{\exp(\beta U_{\min})}{2\alpha(M-1)N} \cdot \ln\frac{1}{2\epsilon}, \quad (55)$$

and

$$t_{\text{mix}}(\epsilon) \leq \frac{2}{\alpha}(M-1)NM^{2N} \cdot \exp(\beta(4U_{\max} - 3U_{\min})) \cdot \left[\ln\frac{1}{2\epsilon} + \frac{1}{2}N\ln M + \frac{1}{2}\beta(U_{\max} - U_{\min})\right]. \quad (56)$$

b) When $0 < \beta < \beta_{\text{th}} = \frac{1}{2(U_{\max}-U_{\min})}\ln(\frac{N+\frac{1}{M-1}}{N-1})$, we have a tighter upper bound

$$t_{\text{mix}}(\epsilon) \leq \frac{\frac{1}{\alpha(M-1)} \cdot \exp(\beta(2U_{\max} - U_{\min})) \cdot \ln\frac{N}{\epsilon}}{N + \frac{1}{M-1} - (N-1)\exp(2\beta(U_{\max} - U_{\min}))}. \quad (57)$$

The proof is relegated to Appendix-E.

Remarks: (i) We observe a phase transition phenomenon for mixing time of our channel-hopping Markov chain. As $\beta \geq \beta_{\text{th}}$, the upper bound of mixing time scales with $\exp(\Omega(N))$; while when $0 < \beta < \beta_{\text{th}}$, the upper bound of mixing time scales with $O(\log(N))$, where N is the number of APs. (ii) By (55)–(57), we see that both the lower bound and the upper bound for mixing time (convergence time) of the channel-hopping Markov chain decreases as α increases. In fact, according to (44), all transition rates of the Markov chain increase as α increases, resulting in fast transitions to the stationary distribution of the channel-hopping Markov chain. Moreover, according to (45), the time interval between two consecutive transitions of the Markov chain also decreases as α increases. Therefore, the mixing time (convergence time) of the channel-hopping Markov chain decreases as α increases.

Overall, we observe a tradeoff between system overhead and mixing time (convergence time) of the channel-hopping Markov chain parameterized by α. As α increases, broadcast message intensity (the number of broadcasting messages per unit time) increases, while the mixing time (convergence time) of the channel-hopping Markov chain decreases. Numerical experiments conducted in Section V-E.5 verify this observation.

TABLE I
NORMALIZED THROUGHPUT AND UTILITY GAP OF THE "WAIT-AND-HOP" ALGORITHM IN A SIX-AP FULL CLIQUE NETWORK

Link No.	1	2	3	4	5	6	ΔU
Wait-and-Hop	0.543	0.543	0.543	0.543	0.542	0.541	-0.00017

D. Evaluation

We evaluate the performance of the proposed "Wait-and-Hop" algorithm through extensive simulations. We set $U_i(\cdot) = \log(\cdot)$ and $\beta = 10$. As the benchmark, the optimal channel assignment state is obtained by exhaustively searching the feasible channel assignments.

1) Simulation Setup: Similar to several prior works [3]–[5], in the simulation we use an idealized CSMA protocol to capture the main features of the protocol while leaving aside some protocol details, including collisions caused by multiple stations counting down to zero in the same time slot and then transmitting together, collisions caused by hidden nodes, ACK, EIFS, and so on.

Typical 802.11b parameter settings are used in the simulation (e.g., $M = 3$). Each AP tries to access the channel according to the standard 802.11 protocol.

In each run of simulations, we gather the statistics of two metrics: (i) normalized aggregate throughput and (ii) normalized system utility. We define ΔT as the ratio of the achieved normalized aggregate throughput and the optimal normalized aggregate throughput. We further define utility gap ΔU as the difference between the normalized system utility achieved and the normalized optimal utility.

2) Aggregate Throughputs and Utilities: We evaluate the achieved normalized aggregate throughputs and the achieved utilities of "Wait-and-Hop" algorithm in networks with different contention graphs.

3) Six-AP Full Clique Network: In a network in which six APs form a clique, it is easy to see that the optimal configuration should be the one in which two APs share a channel. The normalized throughput of each AP and the utility gap of "Wait-and-Hop" are presented in Table I.

As shown in Table I, "Wait-and-Hop" can achieve roughly 99% of the optimal throughput and near optimal utility.

4) Eight-AP Random Networks: We generate ten eight-AP random networks, in which each AP has on average three neighbors in the contention graph. ΔT and ΔU of "Wait-and-Hop" are given in Table II. Averaging over ten networks, we find that the "Wait-and-Hop" algorithm can achieve 99.85% of the optimal aggregate throughput and the average utility gap is 0.00025.

TABLE II
UTILITY GAP AND NORMALIZED THROUGHPUT OF THE "WAIT-AND-HOP" ALGORITHM IN TEN EIGHT-AP RANDOM NETWORKS

Network Number	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Averaged
ΔU	-0.000125	-0.00025	-0.000125	-0.000125	-0.00025	-0.000375	-0.000125	-0.00025	-0.000375	-0.00025	-0.00025
ΔT	99.87%	99.85%	99.90%	99.88%	99.84%	99.80%	99.78%	99.85%	99.85%	99.89%	99.85%

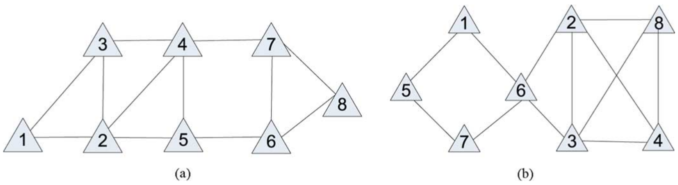

Fig. 4. Two random eight-AP networks.

TABLE III
NORMALIZED THROUGHPUT OF "WAIT-AND-HOP"IN TWO RANDOM EIGHT-AP NETWORKS IN FIG. 4

Link No.	1	2	3	4	5	6	7	8
Network 1	1.00	0.99	1.00	0.99	0.99	0.99	0.99	1.00
Network 2	0.99	0.72	0.72	0.79	0.99	0.99	0.99	0.79

For further details, we show two of these ten networks in Fig. 4. Network 1 in Fig. 4(a) is a three-colorable network. The optimal channel assignment should guarantee that each AP has a normalized throughput of one. Network 2 in Fig. 4(b) has a four-APs clique and hence the contention graph is not three-colorable. The optimal channel assignment state should be such that two of the APs in the four-APs clique share a channel and all the other APs enjoy an unshared channel. One such assignment is $f = [1, 1, 2, 3, 2, 3, 1, 3]$. The achieved normalized throughput of each AP for Networks 1 and 2 are given in Table III.

As shown in Table III, "Wait-and-Hop" can achieve 99.38% and 99.71% of the optimal aggregate throughputs for Networks 1 and 2, respectively. The utility gaps of "Wait-and-Hop" for Networks 1 and 2 are 0.00025 and 0.000125, respectively.

5) Utility Loss: Equation (5) provides a performance loss bound for our Markov approximation. In the worst case, the log–sum–exp approximation can incur a performance loss of $\frac{1}{\beta}\log|\mathcal{F}|$, where $|\mathcal{F}|$ is the number of feasible configurations. In our simulation setup, we have

$$\frac{\log|\mathcal{F}|}{\beta} = \begin{cases} \frac{1}{10}\log 3^6 = 0.6592 \text{ for the 6-AP clique network;} \\ \frac{1}{10}\log 3^8 = 0.8789 \text{ for the 8-AP random network.} \end{cases}$$

Simulation results in Tables I and II show that "Wait-and-Hop" can achieve a utility loss of 0.00017 and 0.00033 for the six-AP full clique networks and eight-AP random networks, respectively.

Comparing the computed performance loss bound with the actual observed utility loss in simulation, we see that the performance loss bound is guaranteed. More importantly, the actual loss can be much smaller than the bound. For all the scenarios tested, "Wait-and-Hop" can actually achieve near-optimal utility.

6) Tradeoff Between Convergence Time and Optimality Gap: We use the total variation distance of probability measures as the

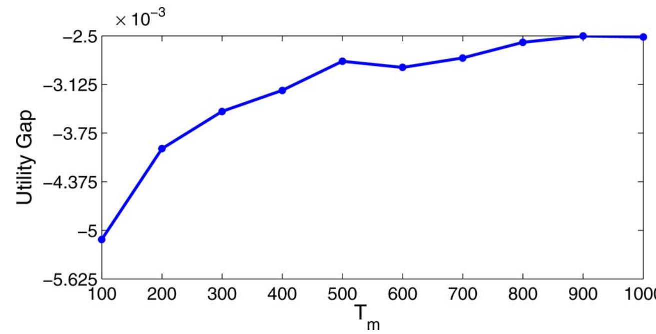

Fig. 5. Mean of utility gap versus T_m with $\alpha = 1$.

metric of our convergence test. For each simulation run, we say the process has converged if

$$\frac{1}{2}\sum_f \left|\hat{p}_f - p_f^*\right| < \sigma, \tag{58}$$

where $\hat{p}_f$ is the measured probability of configuration f in the simulation, p_f^* is the optimal throughput distribution computed by (43), and σ is a threshold. In our simulations, we choose $\sigma = 0.05$.

To scale the transition rate of the process, we put a normalization factor α in the denominator of (45). The larger the α is, the faster the process transits.

The system convergence time, denoted by T_v, consists of two parts: $\Theta \cdot T_m$ and $T_c = \sum_{i=1}^{\Theta} T_i$, in which Θ is the number of channel configuration transitions, T_m is the rate measurement interval, and T_i is the time the process stays for the ith transition. We assume that after each frequency-hopping, the link performs throughput measurement first and the measurement interval is T_m. After T_m, each link sets up its timer and begins to countdown. Thus, we have

$$T_v = \Theta \cdot T_m + \sum_{i=1}^{\Theta} T_i. \tag{59}$$

We first evaluate the case where $\alpha = 1$. Note from (45) that in this case $\sum T_i$ is very small and the system convergence time T_v is dominated by $\Theta \cdot T_m$.

We gather data averaged over ten networks, and T_m is measured in the unit of the mean packet transmission time (i.e., packet duration). Figs. 5 and 6 plot the means of utility gap ΔU and convergence time T_v with respect to T_m, respectively.

In Fig. 6, $\alpha = 1$ and $\sum_i T_i$ is negligible compared to $\Theta \cdot T_m$. We then adjust α so that $E[T_i]$ and T_m are roughly of the same order. Fig. 7 below shows the mean of convergence time with adaptive values of α.

From Figs. 5–7, we can see that the larger the measurement interval T_m is, the closer the achieved utility is to the optimality. The reason is that the accuracy of the rate measurement increases with the measurement interval T_m. On the other hand,

TABLE IV
NORMALIZED THROUGHPUT AND UTILITY GAP OF THE "WAIT-AND-HOP" ALGORITHM IN TEN EIGHT-AP RANDOM NETWORKS WITH $\Theta = 250$

Network Number	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Averaged
ΔU	-0.000575	-0.00205	-0.000725	-0.1493	0.0208	-0.0026	-0.003938	-0.012388	-0.0049	-0.017675	-0.019038
ΔT	99.64%	99.63%	99.65%	88.39%	99.68%	99.74%	99.66%	99.68%	99.70%	99.73%	98.55%

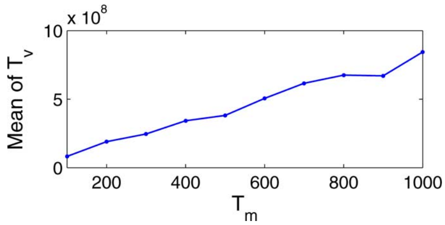

Fig. 6. Mean of convergence time versus T_m with $\alpha = 1$.

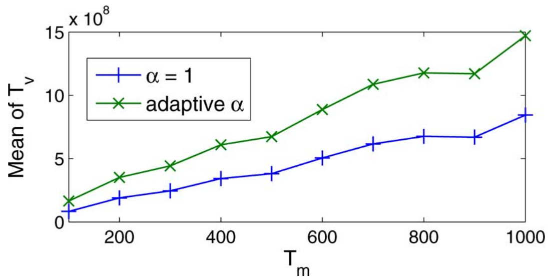

Fig. 7. Mean of convergence time versus T_m with different values of α.

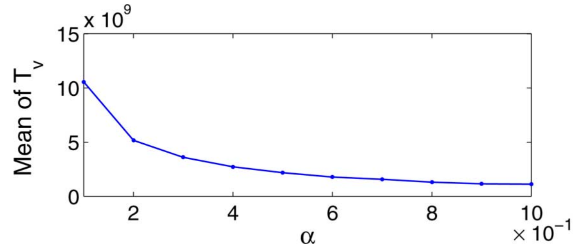

Fig. 8. Mean of convergence time versus α.

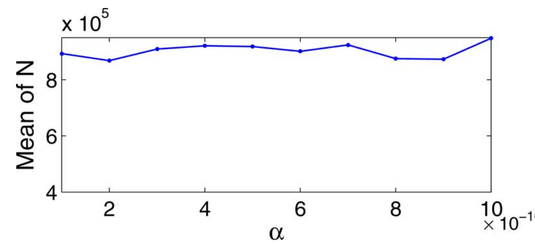

Fig. 9. Mean of the number of transitions versus α.

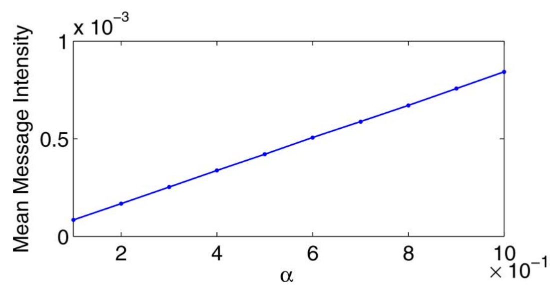

Fig. 10. Mean of the number of transitions versus α.

the convergence time of the process also increases with the measurement interval T_m.

One may find from Figs. 6 and 7 that the "Wait-and-Hop" algorithm requires much time to converge in practice. For example, the convergence time is $10^8/1000\ \mathrm{s} \approx 27.78\ \mathrm{h}$ accordingly to Fig. 7 if we assume the mean packet duration is 1 ms. Our point is that the convergence criterion as in (58) is used for analytical purposes. Indeed, only hundreds of transitions are needed for the "Wait-and-Hop" algorithm to achieve good performance.

To see this, we set up another simulation scenario. We generate ten eight-AP random networks and implement the "Wait-and-Hop" algorithm. The parameters used are as follows: $\alpha = 1$, $\beta = 10$, $T_m = 1000$ and the number of transitions is set to 250. This simulation setting corresponds to a running time of 250 s in real network operation. As shown in Table IV, we find that the "Wait-and-Hop" algorithm can achieve 98.55% of the optimal aggregate throughput. The average utility gap is 0.019038.

7) Tradeoff Between Convergence Time and System Overhead: We examine the tradeoff between the convergence time and the system overhead. By adjusting α, we can scale the rates the process transits, resulting in different $\sum_{i=1}^{\Theta} T_i$. The system overhead (e.g., the number of broadcasting messages) is proportional to the number of channel configuration transitions Θ.

Given a specification of the network, we vary α and repeat the simulation for ten times. The system convergence time and the number of channel configuration transitions are averaged over ten runs. For simplicity, we let $T_m = 0$ first.

Figs. 8 and 9 plot the means of T_v and Θ with respect to α, respectively. The convergence time is measured in the unit of the mean packet duration (e.g., 1 ms in a typical 802.11 network).

Fig. 8 shows that the mean of convergence time decreases with α. Fig. 9 demonstrates that the number of transitions before convergence does not change much with respect to α. This result is consistent with our intuitions: first, as α increases, the process transits faster from a state to another. Hence, the convergence time decreases; second, although the process transits faster, the time it stays in a specific configuration keeps the same ratio, regardless of α. Thus, the mean number of transitions will not be affected much.

On the other hand, as α increases, the number of broadcasting messages per unit time (i.e., the broadcasting message intensity) ρ will increase accordingly. To see this, we plot the mean message intensity ρ with respect to α in Fig. 10.

The tradeoff between convergence time of channel-hopping MC and the system overhead (e.g., broadcasting messages) is then as follows: as α increases, the convergence time of

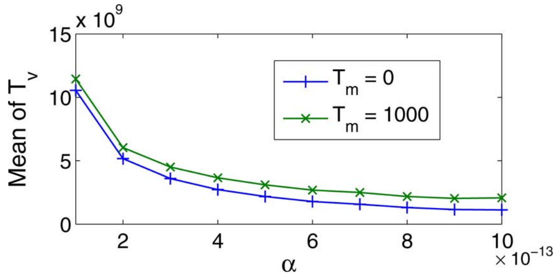

Fig. 11. Mean of convergence time versus α with different values of T_m.

channel-hopping MC decreases while the broadcasting message intensity increases accordingly.

When $T_m \neq 0$, T_v has the similar behavior to that shown in Fig. 8 since $\Theta \cdot T_m$ does not vary much with respect to α. To see this, we plot Fig. 11 in which $T_m = 1000$.

VI. Conclusion

This paper has presented a Markov approximation framework for solving combinatorial network optimization problems. In particular, we show that the log–sum–exp approximation of the optimal objective of a combinatorial problem gives rise to a solution that can be realized by time-reversible Markov chains. These Markov chains usually have desirable structures that can yield distributed algorithms for solving the network optimization problem approximately.

To illustrate our approach, we first apply the Markov approximation technique to the utility maximization problem in CSMA networks. This example offers a fresh perspective to known distributed algorithms. Going beyond, we then show that the Markov approximation technique can help us synthesize new distributed algorithms in new problem domains. We illustrate this by applying the technique to design new distributed algorithms with provable performance for two important practical problems: (i) optimal path selection in multipath transmission and (ii) frequency channel assignment in WLANs. Based on the promising results of our investigation and recent progress documented in [24] and [27]–[31], we believe Markov approximation will find applications in various network optimization problems in many other domains.

Appendix

1) Proof of Lemma 1:

Proof: First, we will construct a continuous-time time-reversible ergodic Markov chain and show that its stationary distribution is $p_f^*(\boldsymbol{x})$ in (11). In particular, we construct a continuous-time Markov chain Y with a finite state space $\mathcal{F}$. We design the Markov chain Y such that any two states f and f' can communicate directly with each other, i.e., the transition rate from f to f' is $q_{f,f'} \neq 0$ for any $f, f' \in \mathcal{F}$. Furthermore, for any $f, f' \in \mathcal{F}$, we set

$$q_{f,f'} = \alpha \left[\exp\left(\beta \sum_{r \in R} x_r(f)\right)\right]^{-1}. \qquad (60)$$

Thus, Y is an ergodic Markov chain with unique stationary distribution. By (60) and (11), we can check that detailed balance equations hold, by Theorems 1.3 and 1.14 in [32], we know that this Y is reversible, and its stationary distribution is indeed $p_f^*(\boldsymbol{x})$ in (11).

Next, we will establish that for any continuous-time time-reversible ergodic Markov chain X, its stationary distribution π can be expressed by the product form $p_f^*(\boldsymbol{x})$ in (11). For the state-transition diagram for the Markov chain X, we map it to an undirected graph $G = (V, E)$, where the node set $V = \mathcal{F}$ is the set of states and any edge $e(i,j) \in E$, $i, j \in V$ represents the state-pair (i,j) with $q_{i,j} \neq 0$.

Let the stationary distribution of state j be denoted by π_j, and transition rate from state j to state j' is denoted by $q_{j,j'}$, then by detailed balance equation of time-reversible Markov chain, we know that $\pi_j q_{j,j'} = \pi_{j'} q_{j',j}$. Let $\rho_{j,j'} = q_{j,j'}/q_{j',j}$ for any $q_{j',j} \neq 0$, then $\pi_{j'} = \pi_j \rho_{j,j'}$.

Since X is an ergodic Markov chain, any two states can reach each other within finite transitions, and G is a connected graph. We can always find a spanning tree to connect all nodes in G and there exists only one path between any pair of nodes. Suppose we have constructed a spanning tree on G. Then, we denote the root state as state 0, and denote nodes in V as state $1, 2, \ldots, |V| - 1$, according to the result of the breadth-first search on the spanning tree. Let PATH(0, i) be the path between state 0 and the state $i (1 \le i \le |V| - 1)$, passing $m_i + 1$ number of states (including states 0 and i). We order the nodes on the path PATH(0,i) according to their distances to state 0, and denoted them as $v_{0,i}^j (0 \le j \le m_i)$. Then, according to detailed balanced equations along the PATH(0, i), we have the following:

$$\pi_i = \pi_0 \cdot \prod_{j=0}^{m_i - 1} \rho_{v_{0,i}^j, v_{0,i}^{j+1}}, \quad 1 \le i \le |V| - 1, \qquad (61)$$

$$\pi_0 + \sum_{i=1}^{|V|-1} \pi_i = 1. \qquad (62)$$

Consequently, we obtain the distribution

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{|V|-1} \prod_{j=0}^{m_k - 1} \rho_{v_{0,k}^j, v_{0,k}^{j+1}}}, \qquad (63)$$

$$\pi_i = \frac{\prod_{j=0}^{m_i - 1} \rho_{v_{0,i}^j, v_{0,i}^{j+1}}}{1 + \sum_{k=1}^{|V|-1} \prod_{j=0}^{m_k - 1} \rho_{v_{0,k}^j, v_{0,k}^{j+1}}}, \quad 1 \le i \le |V| - 1. \qquad (64)$$

We now verify the distribution computed based on the spanning tree, i.e., (63)–(64), is the correct stationary distribution, by testing the detailed balance equations between any two states j, $j' \in V$.

1) If $q_{j,j'} = 0$, then the detailed balance equation trivially holds.
2) If $q_{j,j'} \neq 0$ and the edge $e(j,j')$ belongs to the spanning tree, then by (64), we know that $\pi_{j'} = \pi_j \rho_{j,j'}$, i.e., the detailed balance equation holds.
3) If $q_{j,j'} \neq 0$ and the edge $e(j,j')$ does not belong to the spanning tree, then we focus on the cycle consisting of $PATH(j', j)$ and $e(j,j')$. Starting from node $j' = v_{j',j}^0$,

we can visit nodes $v^k_{j',j}, 1 \leq k \leq m_{j',j} - 1$ and node $j = v^{m_{j',j}}_{j',j}$ in sequence along the $PATH(j', j)$. By Kolmogorov's criteria for time-reversible Markov chain [32], we have

$$\rho_{j',j} = \prod_{k=0}^{m_{j',j}-1} \rho_{v^k_{j',j}, v^{k+1}_{j',j}}. \tag{65}$$

By (64) and (65), we have

$$\frac{\pi_j}{\pi_{j'}} = \prod_{k=0}^{m_{j',j}-1} \rho_{v^k_{j',j}, v^{k+1}_{j',j}} = \rho_{j',j}. \tag{66}$$

Therefore, the detailed balance equation between j and j' holds.

Combining above scenarios, we know that the detailed balance equations between any two states $j, j' \in V$ hold. So the distribution shown in (63)–(64) is indeed the stationary distribution.

Further, the stationary distribution π shown in (63)–(64) can be expressed in the product form in (11) as follows:

$$\pi_0 = \frac{\exp(0)}{\exp(0) + \sum\limits_{k=1}^{|V|-1} \exp\left(\sum\limits_{j=0}^{m_k-1} \log \rho_{v^j_{0,k}, v^{j+1}_{0,k}}\right)}, \tag{67}$$

and for all $1 \leq i \leq |V| - 1$

$$\pi_i = \frac{\exp\left(\sum\limits_{j=0}^{m_i-1} \log \rho_{v^j_{0,i}, v^{j+1}_{0,i}}\right)}{\exp(0) + \sum\limits_{k=1}^{|V|-1} \exp\left(\sum\limits_{j=0}^{m_k-1} \log \rho_{v^j_{0,k}, v^{j+1}_{0,k}}\right)}. \tag{68}$$

So we get the desired conclusion. ■

2) Proof of Theorem 2: Before the further illustration, we need some notation. The vector $\boldsymbol{z}$ and the vector $\boldsymbol{\lambda}$ are updated at time t_m, $m = 1, 2, \ldots$ with $t_0 = 0$. Define $T_m = t_{m+1} - t_m$, and the "period m" as the time between t_m and t_{m+1}, $m = 0, 1, 2, \ldots$ $\boldsymbol{z}(t)$, $\boldsymbol{\lambda}(t)$ remain the same in period m. Let $\boldsymbol{z}(m), \boldsymbol{\lambda}(m)$ be the value of $\boldsymbol{z}(t), \boldsymbol{\lambda}(t)$ for all $t \in [t_m, t_{m+1})$. To begin with, we assume $\boldsymbol{z}(0) = \mathbf{1}$ and $\boldsymbol{\lambda}(0) = \mathbf{0}$ for simplicity. Let $\theta_l(m) = \sum_{l \in f} p_f(\beta \boldsymbol{\lambda}(m))$ denote the link rate for link l in period m. Then, let $\bar{\theta}_l(m)$ be the average link rate measured by link l within period m. For convenience, we also let

$$L_\beta(\boldsymbol{z}, \boldsymbol{\lambda}) = \sum_{s \in S} U_s(z_s) - \sum_{l \in L} \lambda_l \sum_{s: l \in s, s \in S} z_s + \frac{1}{\beta} \log\left[\sum_{f \in \mathcal{F}} \exp\left(\beta \sum_{l \in f} \lambda_l\right)\right]. \tag{69}$$

Then, the stochastic primal-dual algorithm is given as follows: $\forall s \in S$ and $\forall l \in L$,

$$\begin{cases} z_s(m+1) = \left[z_s(m) + \epsilon(m)\left(U'_s(z_s(m)) - \sum_{l: l \in s} \lambda_l(m)\right)\right]^+, \\ \lambda_l(m+1) = \left[\lambda_l(m) - \epsilon(m)\left(\bar{\theta}_l(m) - \sum_{s: l \in s} z_s(m)\right)\right]^+, \end{cases} \tag{70}$$

where $\epsilon(m)$ is the step size and $[\cdot]^+ \triangleq \max(\cdot, 0)$. In general, step size for both source rate and link price updating should be at the same order, though can be different. Here without loss of generality, we use the same step size for both source rate and link price updating.

In the following, we will show that the stochastic primal-dual algorithm converges with probability 1 to the optimal solution of $\mathbf{MP} - \mathbf{MA}$ (24). Thus when $\beta \to \infty$, the stochastic primal-dual algorithm (26) (or (70)) converges with probability 1 to the optimal solution of problem $\mathbf{MP}$ (16).

Now we state the convergence theorem as follows, which is similar to [13, Th. 7].

Theorem 6: Assume that $U'_s(0) < \infty \ \forall s \in S$, $\max_{s,m} z_s(m) < \infty$ and $\max_{l,m} \lambda_l(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:

$$T_{m+1} \geq T_m \ \forall m \geq 1, \tag{71a}$$
$$\epsilon(m) > 0 \ \forall m \geq 1, \tag{71b}$$
$$\sum\nolimits_{m=1}^{\infty} \epsilon(m) = \infty, \tag{71c}$$
$$\sum\nolimits_{m=1}^{\infty} \epsilon^2(m) < \infty, \tag{71d}$$
$$\sum\nolimits_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty. \tag{71e}$$

Then, by running the stochastic primal-dual algorithm (26), $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ converge to $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{\lambda}}$, respectively, with probability 1. Here $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}})$ is the optimal solution to the problem $\mathbf{MP} - \mathbf{MA}$ (24).

It is not hard to see that setting $T_1 = \epsilon(1) = 1, T_m = m$, $\epsilon(m) = \frac{1}{m} \ \forall m \geq 2$ satisfies conditions (71a)–(71e). Further, this setting only depends on the index m, and thus can be generally applied to any network.

Thus by theorem 6, we prove that theorem 2 holds.

Now we state the proof of theorem 6, i.e., the convergence on the stochastic primal-dual algorithm.

Proof: In brief, we prove the convergence by showing that the estimators of stochastic gradients in (70) are unbiased, a standard method of stochastic approximation [15]. The difference between our proof and [13] is that, our proof studies the saddle points of Lagrangian function, while [13] studies the optimal dual solutions directly.

Let $\boldsymbol{y}^0(m)$ be the state of the CSMA Markov chain at the beginning of period m. Define the random vector $U(m) = (\bar{\boldsymbol{\theta}}(m-1), \boldsymbol{z}(m), \boldsymbol{\lambda}(m), \boldsymbol{y}^0(m))$ for $m \geq 1$ and $U(0) = (\boldsymbol{z}(0), \boldsymbol{\lambda}(0), \boldsymbol{y}^0(0))$. For $m \geq 1$, let $\mathcal{F}_m$ be the σ-field generated by $U(0), U(1), \ldots, U(m)$, denoted by

$$\mathcal{F}_m = \sigma(U(0), U(1), \ldots, U(m)). \tag{72}$$

Given $\boldsymbol{z}(m), \boldsymbol{\lambda}(m)$ at the beginning of period m. Let the vector $\boldsymbol{f}(m)$ be the gradient vector of $L_\beta(\boldsymbol{z}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{z}$, and the vector $\boldsymbol{g}(m)$ be the gradient vector of $L_\beta(\boldsymbol{z}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$. Then, we have

$$\begin{cases} f_s(m) = U'_s(z_s(m)) - \sum_{l: l \in s} \lambda_l(m) \ \ \forall s \in S; \\ g_l(m) = \theta_l(m) - \sum_{s: l \in s} z_s(m) \ \ \forall l \in L. \end{cases} \tag{73}$$

However, in stochastic primal-dual algorithm (26), we only have an estimation of $g_l(m)$ for all $l \in L$, denoted by

$$\bar{g}_l(m) = \bar{\theta}_l(m) - \sum_{s:l\in s} z_s(m) \ \forall l \in L. \tag{74}$$

Then $\forall l \in L$, $\bar{g}_l(m)$ can be decomposed into three parts: $\bar{g}_l(m) = g_l(m) + (E[\bar{g}_l(m)|\mathcal{F}_m] - g_l(m)) + (\bar{g}_l(m) - E[\bar{g}_l(m)|\mathcal{F}_m])$.

The first part is the exact gradient $g_l(m)$. The second part is the biased estimation error of $g_l(m)$, denoted by

$$B_l(m) \triangleq E[\bar{g}_l(m)|\mathcal{F}_m] - g_l(m) = E[\bar{\theta}_l(m)|\mathcal{F}_m] - \theta_l(m). \tag{75}$$

The third part is a zero-mean martingale difference noise, denoted by

$$\eta_l(m) \triangleq \bar{g}_l(m) - E[\bar{g}_l(m)|\mathcal{F}_m] = \bar{\theta}_l(m) - E[\bar{\theta}_l(m)|\mathcal{F}_m]. \tag{76}$$

Therefore,

$$\bar{g}_l(m) = g_l(m) + B_l(m) + \eta_l(m) \ \forall l \in L. \tag{77}$$

Recall that $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}})$ is the optimal solution to the problem $\mathbf{MP-MA}$ (24). Thus, $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}})$ is a saddle point for $L_\beta(\boldsymbol{z}, \boldsymbol{\lambda})$.

By using $\|$ to denote the *Euclidean* norm, we define the Lyapunov function $V(\cdot,\cdot)$ as follows:

$$V(\boldsymbol{z}, \boldsymbol{\lambda}) \triangleq \| \boldsymbol{z} - \hat{\boldsymbol{z}} \|^2 + \| \boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}} \|^2 . \tag{78}$$

For any given $\mu > 0$, We also define the set

$$H_\mu \triangleq \{(\boldsymbol{z}, \boldsymbol{\lambda}) : L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}) - L_\beta(\boldsymbol{z}, \hat{\boldsymbol{\lambda}}) \le \mu\}. \tag{79}$$

Since $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}})$ is a saddle point for $L_\beta(\boldsymbol{z}, \boldsymbol{\lambda})$, it follows that

$$L_\beta(\boldsymbol{z}, \hat{\boldsymbol{\lambda}}) \le L_\beta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}}) \le L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}). \tag{80}$$

In the following, we need two steps to establish the convergence result.

1) *Step 1:* we will show that $\forall \mu > 0$, H_μ is recurrent for $\{\boldsymbol{z}(m), \boldsymbol{\lambda}(m)\}$.
2) *Step 2:* we will show that for a sufficient large number m, and any $n \ge m+1$, $\{\boldsymbol{z}(n), \boldsymbol{\lambda}(n)\}$ will reside in H_μ almost surely.

Before the further illustrate of *Step 1* and *Step 2*, we need the following two lemmas. Proofs of them are given at the end of this section.

Lemma 3: $\sum_{m=1}^{\infty} |\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{B}(m)| < \infty$.

Lemma 4: Let $W(n) \triangleq \sum_{i=1}^{n-1} \{\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T \boldsymbol{\eta}(i)\}$, then $W(n)$ converges with probability 1.

Step 1: Since

$$z_s(m+1) = [z_s(m) + \epsilon(m) \cdot f_s(m)]^+ \quad \forall s \in S,$$
$$\lambda_l(m+1) = [\lambda_l(m) - \epsilon(m) \cdot \bar{g}_l(m)]_+ \quad \forall l \in L.$$

By using the fact that the projection $[\cdot]^+$ is nonexpansive [9], we have

$$\begin{aligned}&\| \boldsymbol{z}(m+1) - \hat{\boldsymbol{z}} \|^2 \le \| \boldsymbol{z}(m) + \epsilon(m) \cdot \boldsymbol{f}(m) - \hat{\boldsymbol{z}} \|^2 \\ &= \| \boldsymbol{z}(m) - \hat{\boldsymbol{z}} \|^2 + 2\epsilon(m) \cdot [\boldsymbol{z}(m) - \hat{\boldsymbol{z}}]^T \boldsymbol{f}(m) \\ &+ \epsilon^2(m) \| \boldsymbol{f}(m) \|^2 .\end{aligned}$$

From (77), we also have

$$\begin{aligned}&\| \boldsymbol{\lambda}(m+1) - \hat{\boldsymbol{\lambda}} \|^2 \le \| \boldsymbol{\lambda}(m) - \epsilon(m) \cdot \bar{\boldsymbol{g}}(m) - \hat{\boldsymbol{\lambda}} \|^2 \\ &= \| \boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}} \|^2 - 2\epsilon(m) \cdot [\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}}]^T \bar{\boldsymbol{g}}(m) \\ &+ \epsilon^2(m) \| \bar{\boldsymbol{g}}(m) \|^2 \\ &= \| \boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}} \|^2 - 2\epsilon(m) \cdot [\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}}]^T [\boldsymbol{g}(m) \\ &+ \boldsymbol{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2(m) \| \bar{\boldsymbol{g}}(m) \|^2 .\end{aligned}$$

Since $U_s'(\cdot)$, $z_s(m)$ and $\lambda_l(m)$ are bounded, by (73) and (74), we know that both $\| \boldsymbol{f}(m) \|^2$ and $\| \bar{\boldsymbol{g}}(m) \|^2$ are bounded, we can write that $\| \boldsymbol{f}(m) \|^2 \le C_1$ and $\| \bar{\boldsymbol{g}}(m) \|^2 \le C_2$, where C_1 and C_2 are positive constants. Using this and the above inequalities, we have that

$$\begin{aligned}&V(\boldsymbol{z}(m+1), \boldsymbol{\lambda}(m+1)) \\ &= \| \boldsymbol{z}(m+1) - \hat{\boldsymbol{z}} \|^2 + \| \boldsymbol{\lambda}(m+1) - \hat{\boldsymbol{\lambda}} \|^2 \\ &\le V(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) + 2\epsilon(m) \cdot [(\boldsymbol{z}(m) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(m) - \\ &(\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}})^T \boldsymbol{g}(m)] - 2\epsilon(m) \cdot [\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}}]^T [\boldsymbol{B}(m) + \boldsymbol{\eta}(m)] \\ &+ \epsilon^2(m) \cdot (C_1 + C_2).\end{aligned} \tag{81}$$

Assuming that $(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) \notin H_\mu$ (recall the definition of H_μ in (79)). Then, we have

$$L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}(m)) - L_\beta(\boldsymbol{z}(m), \hat{\boldsymbol{\lambda}}) \ge \mu. \tag{82}$$

Since $L_\beta(\boldsymbol{z}, \boldsymbol{\lambda})$ is concave in $\boldsymbol{z}$ and convex in $\boldsymbol{\lambda}$, $\boldsymbol{f}(m)$ and $\boldsymbol{g}(m)$ are the gradient vectors of $L_\beta(\boldsymbol{z}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, respectively, it follows that

$$L_\beta(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) - L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}(m)) \ge (\boldsymbol{z}(m) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(m), \tag{83}$$
$$L_\beta(\boldsymbol{z}(m), \hat{\boldsymbol{\lambda}}) - L_\beta(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) \ge -(\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}})^T \boldsymbol{g}(m). \tag{84}$$

Summing (83) and (84), and combining (82), we have

$$\begin{aligned}&(\boldsymbol{z}(m) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(m) - (\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}})^T \boldsymbol{g}(m) \\ &\le L_\beta(\boldsymbol{z}(m), \hat{\boldsymbol{\lambda}}) - L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}(m)) \\ &\le -\mu.\end{aligned} \tag{85}$$

Combining the above inequality with (81) we arrive at that

$$\begin{aligned}&V(\boldsymbol{z}(m+1), \boldsymbol{\lambda}(m+1)) \\ &\le V(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) - 2\epsilon(m)\mu + 2\epsilon(m) \cdot \\ &[\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T [\boldsymbol{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2(m) \cdot (C_1 + C_2).\end{aligned} \tag{86}$$

Furthermore,

$$E[V(\boldsymbol{z}(m+1), \boldsymbol{\lambda}(m+1))|\mathcal{F}_m] \\ \leq V(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) - 2\epsilon(m)\mu \\ + 2\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T[\boldsymbol{B}(m)] + \epsilon^2(m) \cdot (C_1 + C_2). \quad (87)$$

By Lemma 3 and conditions (71c) and (71d), we have

$$\left|\sum_m \{\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{B}(m)\}\right| < \infty,$$

and

$$\sum_m \epsilon^2(m) \cdot (C_1 + C_2) < \infty.$$

Then by supermartingale convergence lemma [33], we can conclude that the set H_μ is recurrent for $\{\boldsymbol{z}(m), \boldsymbol{\lambda}(m)\}$.

Step 2: By (81) we have that for $n \geq m+1$,

$$V(\boldsymbol{z}(n), \boldsymbol{\lambda}(n)) \\ \leq V(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) + 2\sum_{i=m}^{n-1}\{\epsilon(i) \cdot [(\boldsymbol{z}(i) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(i) \\ - (\boldsymbol{\lambda}(i) - \hat{\boldsymbol{\lambda}})^T \boldsymbol{g}(i)]\} \\ + 2\sum_{i=m}^{n-1}\{\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T[\boldsymbol{B}(i) + \boldsymbol{\eta}(i)]\} \\ + (C_1 + C_2)\sum_{i=m}^{n-1} \epsilon^2(i). \quad (88)$$

Since $(C_1 + C_2)\sum_{i=1}^{\infty} \epsilon^2(i) < \infty$, $\sum_{i=1}^{\infty} |\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T \boldsymbol{B}(i)| < \infty$ by lemma 3, and $\sum_{i=1}^{\infty} |\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T \boldsymbol{\eta}(i)| < \infty$ by lemma 4, then

$$\lim_{m\to\infty}(C_1 + C_2)\sum_{i=m}^{\infty} \epsilon^2(i) = 0, \quad (89)$$

$$\lim_{m\to\infty}\sum_{i=m}^{\infty} |\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T \boldsymbol{B}(i)| = 0, \quad (90)$$

$$\lim_{m\to\infty}\sum_{i=m}^{\infty} |\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T \boldsymbol{\eta}(i)| = 0. \quad (91)$$

Combining (89)–(91), we know that with probability 1, for any $\zeta > 0$, after $(\boldsymbol{z}(m), \boldsymbol{\lambda}(m))$ returns to H_μ for some sufficiently large m (due to recurrence of H_μ),

$$2\sum_{i=m}^{n-1}\{\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T[\boldsymbol{B}(i) + \boldsymbol{\eta}(i)]\} \\ + (C_1 + C_2)\sum_{i=m}^{n-1} \epsilon^2(i) \leq \zeta \quad (92)$$

for any $n \geq m+1$.

Combining (80) and (85), we have that

$$[(\boldsymbol{z}(i) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(i) - (\boldsymbol{\lambda}(i) - \hat{\boldsymbol{\lambda}})^T \boldsymbol{g}(i)] \leq 0. \quad (93)$$

Therefore, applying (92) and (93) to (88), we have

$$V(\boldsymbol{z}(n), \boldsymbol{\lambda}(n)) \leq V(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) + \zeta \ \forall n \geq m+1.$$

Thus, $(\boldsymbol{z}(n), \boldsymbol{\lambda}(n))$ cannot move far away from H_μ. Since this holds for H_μ with arbitrarily small $\mu > 0$ and any $\zeta > 0$, it follows that $(\boldsymbol{z}, \boldsymbol{\lambda})$ converges to the optimal solution $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}})$ with probability 1. This concludes the proof. ■

Lemma 3: $\sum_{m=1}^{\infty} |\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{B}(m)| < \infty$.

The proof is very similar to the one in [13]. It is done by combining two standard methods in Markov chain: bounds on mixing time [17] and uniformization [16]. We provide the proof here for completeness reason.

Proof: In the following, we consider the period m, i.e., from t_m to t_{m+1}. At time t_m with the transmission aggressiveness vector $\boldsymbol{\lambda}(m)$, denote the corresponding CSMA Markov chain by $Y(t)$. $Y(t)$ is a continuous time Markov chain.

Each state $\boldsymbol{y}$ is a $|L|$-dimensional vector, with lth element $y_l \in \{0, 1\}$ denote the capacity of link l at state $\boldsymbol{y}$ $\forall l \in L$. The number of states is $|Y| \leq 2^{|L|}$.

By (21) $\forall \boldsymbol{y}$, the stationary distribution of state $\boldsymbol{y}$ is

$$\pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m)) = p_{\boldsymbol{y}}(\beta\boldsymbol{\lambda}) = \frac{\exp\left(\beta\sum_{l\in L} y_l\,\lambda_l\right)}{\sum_{\boldsymbol{y}'}\exp\left(\beta\sum_{l\in L} y'_l\,\lambda_l\right)} \\ = \frac{\exp\left(\beta\sum_{l\in L} y_l\,\lambda_l\right)}{C(\boldsymbol{\lambda}(m))}, \quad (94)$$

where $C(\boldsymbol{\lambda}(m)) = \sum_{\boldsymbol{y}'}\exp\left(\beta\sum_{l\in L} y'_l\,\lambda_l\right)$.

Since $\boldsymbol{\lambda}(m) \geq \boldsymbol{0}$,

$$C(\boldsymbol{\lambda}(m)) \leq \sum_{\boldsymbol{y}'}\exp(\beta\boldsymbol{1}^T\boldsymbol{\lambda}(m)) \leq 2^{|L|}\exp(\beta\boldsymbol{1}^T\boldsymbol{\lambda}(m)).$$

Thus, the minimum probability in the stationary distribution

$$\pi_{\min}(\boldsymbol{\lambda}(m)) \triangleq \min_{\boldsymbol{y}} \pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m)) \\ \geq \frac{1}{C(\boldsymbol{\lambda}(m))} = \exp(-|L|\cdot\log 2 - \beta\boldsymbol{1}^T\boldsymbol{\lambda}(m)).$$

Since $\lambda_{\max} = \max_{l,m}\lambda_l(m) < \infty$, we have

$$\pi_{\min}(\boldsymbol{\lambda}(m)) \geq \exp(-|L|\cdot\log 2 - \beta|L|\lambda_{\max}) \\ = \exp(-|L|\cdot(\log 2 + \beta\lambda_{\max})). \quad (95)$$

To utilize the existing bounds on convergence to the stationary distribution of discrete-time Markov chain, we uniformize the continuous-time Markov chain $Y(t)$. Uniformization [16] plays the role of bridge between discrete-time Markov chain and continuous-time Markov chain.

Let the transition rate matrix of $Y(t)$ is denoted by $Q = \{Q(\boldsymbol{y}, \boldsymbol{y}')\}$. Construct a discrete-time Markov chain $Z(n)$ with its probability transition matrix $P = I + Q/v_m$, where I is the identity matrix. Then, consider a system that successive states visited form a Markov chain $Z(n)$ and the times at which the system changes its state form a Poisson process $N(t)$. Here, $N(t)$ is an independent Poisson process with rate v_m. Then the state of this system at time t is denoted by $Z(N(t))$, which is called a *subordinated Markov chain*.

Let

$$v_m = |L|\cdot\exp(\beta\lambda_{\max}). \quad (96)$$

Since $\forall \boldsymbol{y}, \boldsymbol{y}'$, $Q(\boldsymbol{y}, \boldsymbol{y}') \leq \exp(\beta\lambda_l(m)) \leq \exp(\beta\lambda_{\max})$, and $\boldsymbol{y}$ can at most transit to $|L|$ other states, thus $\Sigma_{\boldsymbol{y}\neq\boldsymbol{y}'}Q(\boldsymbol{y}, \boldsymbol{y}') \leq |L|\cdot\exp(\beta\lambda_{\max}) = v_m$. Then by uniformization theorem [16], $Y(t)$ and $Z(N(t))$ has the same distribution, denoted by $Y(t) \overset{d}{=} Z(N(t))$.

Now let the vector $\boldsymbol{\omega}_m(t) = \{\omega_m(t, \boldsymbol{y})\}$ be the probabilities of all states at time $t_m + t(0 \leq t \leq T_m)$, given that the initial state at time t_m is $\boldsymbol{y}^0(m)$ and that the transmission aggressiveness during period $m([t_m, t_{m+1}))$ are $\boldsymbol{\lambda}(m)$. Let $\boldsymbol{y}(t_m+t)$ be the state at time t_m+t, then

$$\begin{aligned}
&E[\bar{\theta}_l(m)|\mathcal{F}_m]] \\
&= E\left[\int_0^{T_m} 1\cdot I(y_l(t_m+t)=1)dt/T_m\right] \\
&= \int_0^{T_m} 1\cdot P(y_l(t_m+t)=1)dt/T_m \\
&= \int_0^{T_m} E[y_l(t_m+t)|\mathcal{F}_m]dt/T_m \\
&= \int_0^{T_m} \sum_{\boldsymbol{y}'} y'\cdot \omega_m(t,\boldsymbol{y}')dt/T_m \\
&= \sum_{\boldsymbol{y}'} y'\cdot \int_0^{T_m} \omega_m(t,\boldsymbol{y}')dt/T_m \\
&= \sum_{\boldsymbol{y}'} y'\cdot \bar{\omega}_m(\boldsymbol{y}'),
\end{aligned} \tag{97}$$

where $\bar{\omega}_m(\boldsymbol{y}') = \int_0^{T_m}\omega_m(t,\boldsymbol{y}')dt/T_m$ is the time-averaged probability of state y' in the interval.

Since the initial distribution is concentrated at a single definite starting state $\boldsymbol{y}^0(m)$, we denote this distribution by $\delta_{\boldsymbol{y}^0}$. We let $\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))$ be the probability of $\boldsymbol{y}^0(m)$ in the stationary distribution of $Y(t)$. Let $\boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m})) \triangleq \{\pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m)\}$ be the stationary distribution of $Y(t)$, then by uniformization theorem [16], $\boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))$ is also the stationary distribution of $Z(n)$.

We use $\|\cdot\|_{TV}$ to denote the total variation distance between two distributions [17], which satisfies triangle inequality. We use ρ_2 to denote the second largest eigenvalue of transition matrix P. Thus for reversible discrete-time Markov chain $Z(n)$ with transition matrix P, and for any $n\ge 0$, we have the following inequality [17]:

$$\|\delta_{\boldsymbol{y}^0}P^n - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\|_{TV} \le \frac{1}{2}\sqrt{\frac{1-\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))}{\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))}}\cdot\rho_2^n.$$

Therefore,

$$\begin{aligned}
&\|\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\|_{TV} \\
&= \left\|\sum_{n=0}^{\infty}\frac{(v_m t)^n}{n!}\exp(-v_m t)\delta_{\boldsymbol{y}^0}P^n - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\right\|_{TV} \\
&\le \sum_{n=0}^{\infty}\frac{(v_m t)^n}{n!}\exp(-v_m t)\left\|\delta_{\boldsymbol{y}^0}P^n - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\right\|_{TV} \\
&\le \frac{1}{2}\sqrt{\frac{1-\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))}{\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))}}\cdot\sum_{n=0}^{\infty}\frac{(v_m t\rho_2)^n}{n!}\exp(-v_m t) \\
&= \frac{1}{2}\sqrt{\frac{1-\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))}{\pi_{\boldsymbol{y}^0}(\boldsymbol{\lambda}(m))}}\cdot\exp(-v_m(1-\rho_2)t) \\
&\le \frac{1}{2}\sqrt{\frac{1}{\pi_{\min}(\boldsymbol{\lambda}(m))}}\cdot\exp(-v_m(1-\rho_2)t).
\end{aligned}$$

Furthermore,

$$\begin{aligned}
&\|\bar{\omega}_m - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\|_{TV} \\
&= \|\int_0^{T_m}[\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))]dt/T_m\|_{TV} \\
&\le \int_0^{T_m}\|\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\|_{TV}dt/T_m \\
&\le \frac{1}{2}\sqrt{\frac{1}{\pi_{\min}(\boldsymbol{\lambda}(m))}}\frac{1}{v_m(1-\rho_2)T_m}.
\end{aligned} \tag{98}$$

Now we bound ρ_2 by Cheeger's inequality [17]:

$$\rho_2 \le 1-\phi^2/2$$

where ϕ is the "Conductance" of P, defined as

$$\phi \triangleq \min_{N\subset\Omega,\pi(N)\in(0,1/2]}\frac{F(N,N^c)}{\pi_N(\boldsymbol{\lambda}(m))}.$$

Here, Ω is the state space,

$$\pi_N(\boldsymbol{\lambda}(m)) = \Sigma_{\boldsymbol{y}\in N}\,\pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m)),$$

and

$$F(N,N^c) = \Sigma_{\boldsymbol{y}\in N,\boldsymbol{y}'\in N^c}\,\pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m))P(\boldsymbol{y},\boldsymbol{y}').$$

Thus, we have

$$\begin{aligned}
\phi &\ge \min_{N\subset\Omega,\pi(N)\in(0,1/2]}F(N,N^c) \\
&\ge \min_{\boldsymbol{y}\ne\boldsymbol{y}',P(\boldsymbol{y},\boldsymbol{y}')>0}F(\boldsymbol{y},\boldsymbol{y}') \\
&= \min_{\boldsymbol{y}\ne\boldsymbol{y}',P(\boldsymbol{y},\boldsymbol{y}')>0}\pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m))P(\boldsymbol{y},\boldsymbol{y}') \\
&\ge \min_{\boldsymbol{y}}\pi_{\boldsymbol{y}}(\boldsymbol{\lambda}(m))/v_m \\
&= \pi_{\min}(\boldsymbol{\lambda}(m))/v_m,
\end{aligned}$$

and then

$$\frac{1}{1-\rho_2} \le \frac{2}{\phi^2} = 2\cdot v_m^2[\pi_{\min}(\boldsymbol{\lambda}(m))]^{-2}. \tag{99}$$

Combined (99), (95), (96) with (98), it follows that

$$\begin{aligned}
\|\bar{\omega}_m - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))\|_{TV} &\le \frac{v_m}{T_m}[\pi_{\min}(\boldsymbol{\lambda}(m))]^{-5/2} \\
&= |L|\cdot\tau/T_m,
\end{aligned} \tag{100}$$

where $\tau \triangleq \exp[(5/2|L|+1)\beta\lambda_{\max} + 5/2|L|\log 2]$.

So by (75) and (97), we have $\forall l\in L$,

$$\begin{aligned}
|B_l(m)| &= |E[\bar{\theta}_l(m)|\mathcal{F}_m] - \theta_l(m)| \\
&= |\sum_{\boldsymbol{y}'}y'\cdot\bar{\omega}_m(\boldsymbol{y}') - \sum_{\boldsymbol{y}'}y'\cdot\pi_{\boldsymbol{y}'}(\boldsymbol{\lambda}(m)))| \\
&\le 2\cdot||\bar{\omega}_m - \boldsymbol{\pi}(\boldsymbol{\lambda}(\boldsymbol{m}))||_{TV} \\
&\le 2|L|\cdot\tau/T_m.
\end{aligned}$$

Since $\forall l \in L$, $\hat{\lambda}_l$ is bounded and $\hat{\lambda}_l < \bar{r}$ for some $\bar{r} > 0$, then we have

$$|[\hat{\lambda}_l - \lambda_l(m)]| \leq \bar{r} + \lambda_{\max} \quad \forall l \in L.$$

Therefore,

$$\begin{aligned}
&\sum\nolimits_{m=1}^{\infty} |\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{B}(m)| \\
&\leq 2|L|^2 \sum\nolimits_{m=1}^{\infty} \epsilon(m) \cdot [\bar{r} + \lambda_{\max}] \cdot \tau / T_m \\
&= 2|L|^2 [\bar{r} + \lambda_{\max}] \tau \sum\nolimits_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} \\
&< \infty,
\end{aligned}$$

where the last step follows from condition (71c). ■

Lemma 4: Let $W(n) \triangleq \sum_{i=1}^{n-1} \{\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(i)]^T \boldsymbol{\eta}(i)\}$, then $W(n)$ converges with probability 1.

Proof: First, we prove that $W(n)$ is a martingale. By (72) and (76), we know that $\boldsymbol{\eta}(n-1) \in \mathcal{F}_n$, $E[\boldsymbol{\eta}(n-1)|\mathcal{F}_{n-1}] = \mathbf{0}$. Further, $\forall l \in L$, $|\eta_l(n)|$ is bounded and $|\eta_l(n)| < c_3$ for some $c_3 > 0$. Thus, $W(n) \in \mathcal{F}_n$, $E|W(n)| < \infty$ $\forall n$ and $E(W(n)|\mathcal{F}_{n-1}) - W(n-1) = \epsilon(n-1) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(n-1)]^T E[\boldsymbol{\eta}(n-1)|\mathcal{F}_{n-1}] = 0$.

Then, we prove that $\sup_n E(W(n)^2) < \infty$.

Since $\forall l \in L$, $\hat{\lambda}_l$ is bounded and $\hat{\lambda}_l < \bar{r}$ for some $\bar{r} > 0$, then we have

$$|[\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{\eta}(m)| \leq |L| \cdot c_3 [\bar{r} + \lambda_{\max}].$$

Thus,

$$\begin{aligned}
&\sup_n E(W(n)^2) \\
&= \sup \sum\nolimits_{m=1}^{n-1} E\{[\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{\eta}(m)]^2\} \\
&\leq \sum\nolimits_{m=1}^{\infty} E\{[\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}(m)]^T \boldsymbol{\eta}(m)]^2\} \\
&\leq \sum\nolimits_{m=1}^{\infty} \{\epsilon(m)^2 |L|^2 c_3^2 [\bar{r} + \lambda_{\max}]^2\} \\
&= |L|^2 c_3^2 [\bar{r} + \lambda_{\max}]^2 \sum\nolimits_{m=1}^{\infty} \{\epsilon(m)^2\} \\
&< \infty,
\end{aligned}$$

where the last step follows from condition (71b). By Martingale convergence theorem [15], $W(n)$ converges with probability 1. ■

3) Proof of Proposition 2:

Proof: In the "Wait-and-Hop" algorithm, the state sojourn time is exponentially distributed and the transition probability is independent of time t, so the state transition process forms a homogeneous continuous-time Markov chain.

Let $\xi_{f,f'}$ be the probability that the process will enter state f' when it leaves state f upon count-down expiration. Let $\mathcal{N}(f)$ be the set of states which are directly connected to state f. In the "Wait-and-Hop" algorithm, the next state of f has equal probability to be any state f' where $f' \in \mathcal{N}(f)$. Specifically, noticing $|\mathcal{N}(f)| = (M-1)N$, we have

$$\xi_{f,f'} = \frac{1}{|\mathcal{N}(f)|} = \frac{1}{(M-1)N} \quad \forall f' \in \mathcal{N}(f). \tag{101}$$

In the following, we show the "Wait-and-Hop" implementation realizes a channel-hopping Markov chain with transition rate shown in (44).

1) First, all the transition rates of the channel-hopping process are finite.
2) Second, it is not hard to see that all channel assignment configurations can reach each other within a finite number of transitions. Thus, the constructed Markov chain is irreducible.
3) Third, the detailed balance equation holds between any two adjacent states. In fact, given the current state f, since each AP counts down with a rate

$$\alpha(M-1)\left(\exp\left(\beta \cdot x_f\right)\right)^{-1},$$

then the process leaves state f with a rate

$$N\alpha(M-1)\left(\exp\left(\beta \cdot x_f\right)\right)^{-1}.$$

With probability $\xi_{f,f'} = \frac{1}{(M-1)N}$, the process jumps to an adjacent state f' when leaving state f. Hence, we can calculate the transition rate from state f to state f' as follows:

$$\begin{aligned}
&\frac{1}{(M-1)N} \times \alpha N(M-1)\left(\exp\left(\beta \cdot x_f\right)\right)^{-1} \\
&\qquad = \alpha\left(\exp\left(\beta \cdot x_f\right)\right)^{-1}.
\end{aligned} \tag{102}$$

By (102) and (43), we obtain $p_f^* q_{f,f'} = p_{f'}^* q_{f',f}$. According to [32, Th. 1.2], the constructed Markov Chain is time-reversible and its stationary distribution is indeed (43). ■

4) Proof of Theorem 4: We adopt the quantization error model proposed in [24]. For each configuration $f \in \mathcal{F}$ with normalized system utility x_f, we denote the measured normalized system utility as $\bar{x}_f$. Then, the perturbation error is defined as

$$\varepsilon_f \triangleq \bar{x}_f - x_f \; \forall f \in \mathcal{F}.$$

Without loss of generality, we assume ε_f belongs to the bounded region $[-\Delta_f, 0]$ for any $f \in \mathcal{F}$. We quantize such error ε_f into $n_f + 1$ values $[-\Delta_f, \ldots, -\frac{\Delta_f}{n_f}, 0]$. We also assume that $\varepsilon_f = -\frac{j\Delta_f}{n_f}$ with probability $\eta_{f_j}, j \in \{0, \ldots, n_f\}$ and $\sum_j \eta_{f_j} = 1$. We will see later that our results remain the same with or without the quantization. The difference between our setting and the setting in [24] is the following: Δ_f in our setting is time dependent while Δ_f in [24] is time independent.

Following the same perturbation analysis, we obtain the stationary distribution of perturbed channel-hopping Markov chain as follows:

$$\bar{p}_f = \frac{\sigma_f \exp(\beta x_f)}{\sum\limits_{f' \in \mathcal{F}} \sigma_{f'} \exp(\beta x_{f'})} \; \forall f \in \mathcal{F},$$

where

$$\begin{aligned}
\sigma_f &\triangleq \sum_{j=0}^{n_f} \eta_{f_j} \cdot \exp(-\beta \frac{j}{n_f} \Delta_f) && (103) \\
&= E[\exp(\beta \varepsilon_f)]. && (104)
\end{aligned}$$

We omit details here since it is very similar to [24].

Now, we already know

$$p_f^* = \frac{\exp(\beta x_f)}{\sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})} \; \forall f \in \mathcal{F}.$$

To proceed, let

$$\bar{\sigma} \triangleq \frac{\sum_{f' \in \mathcal{F}} \sigma_{f'} \exp(\beta x_{f'})}{\sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})}. \tag{105}$$

Then, it is straightforward to verify that $\frac{p_f^*}{\bar{p}_f} = \frac{\bar{\sigma}}{\sigma_f}$. Consequently, we have

$$p_f^* \geq \bar{p}_f, \text{ if and only if } \sigma_f \leq \bar{\sigma}. \tag{106}$$

We know that total variation distance can be expressed as

$$\begin{aligned} \|\boldsymbol{p^*} - \bar{\boldsymbol{p}}\|_{TV} &= \frac{1}{2} \sum_{f \in \mathcal{F}} |p_f^* - \bar{p}_f| \\ &= \sum_{f \in A} (p_f^* - \bar{p}_f), \end{aligned}$$

where $A \triangleq \{f \in \mathcal{F} : p_f^* \geq \bar{p}_f\}$. By (106), we know $A = \{f \in \mathcal{F} : \sigma_f \leq \bar{\sigma}\} \subset \mathcal{F}$.

Therefore, $\forall f \in A$,

$$\begin{aligned} p_f^* - \bar{p}_f &= \frac{\exp(\beta x_f)}{\sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})} - \frac{\sigma_f \exp(\beta x_f)}{\sum_{f' \in \mathcal{F}} \sigma_{f'} \exp(\beta x_{f'})} \\ &= \frac{\exp(\beta x_f)}{\sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})} - \frac{\sigma_f \exp(\beta x_f)}{\bar{\sigma} \sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})} \\ &= \frac{\exp(\beta x_f)}{\sum_{f' \in \mathcal{F}} \exp(\beta x_{f'})} \left[1 - \frac{\sigma_f}{\bar{\sigma}}\right]. \end{aligned} \tag{107}$$

By (103), we know that $\forall f \in \mathcal{F}$

$$\sigma_f \leq \sum_{j=0}^{n_f} \eta_{f_j} \cdot 1 = 1. \tag{108}$$

On the other hand, by (104) and Jensen's inequality [9], we know that $\forall f \in \mathcal{F}$

$$\begin{aligned} \sigma_f &= E[\exp(\beta \varepsilon_f)] \\ &\geq \exp(E(\beta \varepsilon_f)) \\ &\geq \exp(-\beta |E(\varepsilon_f)|). \end{aligned}$$

Then, we need the following lemma. Proofs of it is given at the end of this section.

Lemma 5: For any $f \in \mathcal{F}$,

$$\begin{aligned} |E(\varepsilon_f)| &\leq \frac{2\phi U_{\max}}{T_m}, \\ \sigma_f &\geq \exp(-\beta \frac{2\phi U_{\max}}{T_m}), \end{aligned}$$

where $\phi = L \cdot [(\frac{5}{2} L(4 + \log 2) + 4]$ and L is the number of links in underlying CSMA network.

By (105) and (108), we have $\bar{\sigma} \leq 1$; hence,

$$1 - \frac{\sigma_f}{\bar{\sigma}} \leq 1 - \exp(-\beta \frac{2\phi U_{\max}}{T_m}) \; \forall f \in A \subset \mathcal{F}.$$

Thus by (107), we have $\forall f \in A$,

$$\begin{aligned} p_f^* - \bar{p}_f &= \frac{\exp(\beta x_f)}{\sum\limits_{f' \in \mathcal{F}} \exp(\beta x_{f'})} [1 - \frac{\sigma_f}{\bar{\sigma}}] \\ &\leq \frac{\exp(\beta x_f)}{\sum\limits_{f' \in \mathcal{F}} \exp(\beta x_{f'})} (1 - \exp(-\beta \frac{2\phi U_{\max}}{T_m})). \end{aligned}$$

As a result, we have

$$\begin{aligned} \|\boldsymbol{p^*} - \bar{\boldsymbol{p}}\|_{TV} &= \sum_{f \in A} (p_f^* - \bar{p}_f) \\ &\leq \sum_{f \in A} \frac{\exp(\beta x_f)}{\sum\limits_{f' \in \mathcal{F}} \exp(\beta x_{f'})} (1 - \exp(-\beta \frac{2\phi U_{\max}}{T_m})) \\ &\leq \sum_{f \in \mathcal{F}} \frac{\exp(\beta x_f)}{\sum\limits_{f' \in \mathcal{F}} \exp(\beta x_{f'})} (1 - \exp(-\beta \frac{2\phi U_{\max}}{T_m})) \\ &= 1 - \exp\left(-\beta \frac{2\phi U_{\max}}{T_m}\right). \end{aligned}$$

We conclude the proof by bounding the performance gap as follows:

$$\begin{aligned} |\boldsymbol{p^*} \boldsymbol{x}^T - \bar{\boldsymbol{p}} \boldsymbol{x}^T| &= |\sum_{f \in \mathcal{F}} (p_f^* - \bar{p}_f) x_f| \\ &\leq U_{\max} \sum_{f \in \mathcal{F}} |p_f^* - \bar{p}_f| \\ &= 2U_{\max} \|\boldsymbol{p^*} - \bar{\boldsymbol{p}}\|_{TV} \\ &\leq 2U_{\max} (1 - \exp(-\beta \frac{2\phi U_{\max}}{T_m})). \end{aligned}$$

Lemma 5: For any $f \in \mathcal{F}$,

$$\begin{aligned} |E(\varepsilon_f)| &\leq \frac{2\phi U_{\max}}{T_m}, \\ \sigma_f &\geq \exp(-\beta \frac{2\phi U_{\max}}{T_m}), \end{aligned}$$

where $\phi = L \cdot [(\frac{5}{2} L(4 + \log 2) + 4]$ and L is the number of links in underlying CSMA network.

Proof: Note that there are two Markov chains interacting with each other in the whole system, with one running on top of the other. The one on the top is the channel-hopping Markov chain with its states being channel assignment configurations $f \in \mathcal{F}$. The one below it is the CSMA Markov chain with its states being link independent sets $\omega \in \Omega$. Here we denote the state space for the underlying CSMA Markov chain as Ω. The stationary distribution for underlying CSMA Markov chain under channel configuration f is denoted by $\boldsymbol{\pi}_f = (\pi_{f,\omega}, \omega \in \Omega)$. The state distribution of CSMA Markov chain under channel configuration f at time t is denoted by $\boldsymbol{\eta}_f(t) = (\eta_{f,\omega}(t), \omega \in \Omega)$. Normalized system utility under channel configuration f and link independent set ω is denoted by $y_{f,\omega}$.

For any $f \in \mathcal{F}$, we have $E(\varepsilon_f) = E(\bar{x}_f - x_f) = E(\bar{x}_f) - E(x_f)$. In a way similar to how we derive (97), we obtain $E(\bar{x}_f) = \sum_\omega \bar{\eta}_{f,\omega} \cdot y_{f,\omega}$, where $\bar{\eta}_{f,\omega} = \int_0^{T_m} \eta_{f,\omega}(t)dt/T_m$. Meanwhile, we also have $E(x_f) = \sum_\omega \pi_{f,\omega} \cdot y_{f,\omega}$. Thus,

$$\begin{aligned}|E(\varepsilon_f)| &= |E(\bar{x}_f) - E(x_f)| \\ &= |\sum_\omega (\bar{\eta}_{f,\omega} - \pi_{f,\omega}) \cdot y_{f,\omega}| \\ &\le 2U_{\max} \|\bar{\boldsymbol{\eta}}_f - \pi_f\|_{TV},\end{aligned}$$

where $\bar{\boldsymbol{\eta}}_f = (\bar{\eta}_{f,\omega}, \omega \in \Omega)$.

In a way similar to how we derive (100), we have

$$\|\bar{\boldsymbol{\eta}}_f - \pi_f\|_{TV} \le \frac{L \cdot \tau'}{T_m},$$

where L is the number of links in underlying CSMA network and $\tau' = \exp[(\frac{5}{2}L + 1) \cdot \beta' \lambda'_{\max} + \frac{5}{2}L \cdot \log 2]$. Here $\beta' \lambda'_{\max}$ is related to the protocol used by underlying CSMA networks. Since we adopt 802.11b protocol, we have $\beta' \lambda'_{\max} \le 4$. Thus, $\tau' \le [\frac{5}{2}L(4 + \log 2) + 4]$.

Denote ϕ as $L \cdot [\frac{5}{2}L(4 + \log 2) + 4]$, we have

$$\|\bar{\boldsymbol{\eta}}_f - \pi_f\|_{TV} \le \frac{\phi}{T_m}.$$

Thus, we have

$$\begin{aligned}|E(\varepsilon_f)| &\le \frac{2\phi U_{\max}}{T_m}, \\ \sigma_f &\ge \exp(-\beta \frac{2\phi U_{\max}}{T_m}).\end{aligned}$$

This concludes the proof of Lemma 5. ■

5) Proof of Theorem 5: **(a)** We present the proof for the lower bound of mixing time (55), which is based on the spectral analysis method [17], [25].

The channel-hopping Markov chain is a *continuous-time* Markov chain and its stationary distribution is

$$p_f^* = \frac{\exp[\beta x_f]}{\sum_{f' \in \mathcal{F}} \exp[\beta x_{f'}]} \; \forall f \in \mathcal{F}.$$

Since $\sum_{f' \in \mathcal{F}} \exp[\beta x_{f'}] \le |\mathcal{F}| \exp(\beta U_{\max})$ and $|\mathcal{F}| = M^N$, the minimum probability in the stationary distribution

$$p_{\min} \triangleq \min_{f \in \mathcal{F}} p_f^* \ge \frac{\exp(\beta U_{\min})}{|\mathcal{F}| \cdot \exp(\beta U_{\max})} \tag{109}$$

$$= \frac{1}{M^N} \exp(-\beta(U_{\max} - U_{\min})). \tag{110}$$

Similar to the proof of Lemma 3, we utilize the uniformization technique. Denote $Q = \{q_{f,f'}\}$ as the transition rate matrix of perfect Markov chain. Construct a discrete-time Markov chain $Z(n)$ with its probability transition matrix $P = I + \frac{Q}{\theta}$, where I is the identity matrix. Let ρ_2 denotes the second largest eigenvalue of transition matrix P for Markov chain $Z(n)$.

Note that since $\forall f, f' \in \mathcal{F}$,

$$q_{f,f'} \le \alpha \cdot \exp(-\beta x_f) \le \alpha \exp(-\beta U_{\min}),$$

and f can at most transit to $(M-1)N$ other states, thus $\sum_{f' \ne f} q_{f,f'} \le (M-1)N\alpha \exp(-\beta U_{\min})$.

Thus, the uniformization constant is given as follows:

$$\theta = (M-1)N\alpha \exp(-\beta U_{\min}). \tag{111}$$

By uniformization theorem and spectral gap inequality [17], [25], we have

$$\begin{aligned}\frac{\exp(-\theta(1-\rho_2)t)}{2} &\le \max_{f \in \mathcal{F}} \|\boldsymbol{H}_t(f) - \boldsymbol{p}^*\|_{TV} \\ &\le \frac{\exp(-\theta(1-\rho_2)t)}{2\sqrt{p_{\min}}}.\end{aligned}$$

Therefore,

$$\frac{1}{\theta(1-\rho_2)} \ln \frac{1}{2\epsilon} \le t_{\text{mix}}(\epsilon) \tag{112}$$

$$\le \frac{1}{\theta(1-\rho_2)} \left[\ln \frac{1}{2\epsilon} + \frac{1}{2} \ln \frac{1}{p_{\min}} \right]. \tag{113}$$

Now we bound ρ_2 by Cheeger's inequality [17], [25]:

$$1 - 2\Phi \le \rho_2 \le 1 - \frac{1}{2}\Phi^2, \tag{114}$$

where Φ is the "Conductance" of P, defined as

$$\Phi \triangleq \min_{N \subset \mathcal{F}, \pi_N \in (0,1/2]} \frac{F(N, N^c)}{\pi_N}. \tag{115}$$

Here,

$$\begin{aligned}\pi_N &= \sum_{f \in N} p_f^*, \\ F(N, N^c) &= \sum_{f \in N, f' \in N^c} p_f^* P(f, f').\end{aligned}$$

Combining (112) and (114), we have

$$\frac{1}{2\theta\Phi} \ln \frac{1}{2\epsilon} \le t_{\text{mix}}(\epsilon) \le \frac{2}{\theta\Phi^2} [\ln \frac{1}{2\epsilon} + \frac{1}{2} \ln \frac{1}{p_{\min}}]. \tag{116}$$

First, we give an upper bound of Φ. For any $N' \subset \mathcal{F}$, $\pi(N') \in (0, 1/2]$,

$$\begin{aligned}\Phi &= \min_{N \subset \mathcal{F}, \pi(N) \in (0,1/2]} \frac{F(N, N^c)}{\pi_N} \\ &\le \frac{1}{\pi_{N'}} \sum_{f \in N', f' \in N'^c} p_f^* P(f, f') \\ &= \frac{1}{\pi_{N'}} \sum_{f \in N'} p_f^* \cdot \left(\sum_{f' \in N'^c} P(f, f') \right) \\ &\le \frac{1}{\pi_{N'}} \sum_{f \in N'} p_f^* \\ &= 1.\end{aligned} \tag{117}$$

Combining (111), (116), and (117), we have the lower bound of $t_{\rm mix}(\epsilon)$:

$$t_{\rm mix}(\epsilon) \geq \frac{1}{2\theta}\ln\frac{1}{2\epsilon} = \frac{\exp(\beta U_{\min})}{2\alpha(M-1)N}\cdot \ln\frac{1}{2\epsilon}.$$

Now we give a lower bound of Φ. When $q_{f,f'} \neq 0 \ \forall \in \mathcal{F}$, by (44),

$$q_{f,f'} = \alpha\exp(-\beta x_f) \geq \alpha \exp(-\beta U_{\max}). \tag{118}$$

Combining (115) and (118), we have

$$\begin{aligned}\Phi &\geq \min_{N\subset\mathcal{F},\pi(N)\in(0,1/2]} F(N,N^c)\\ &\geq \min_{f\neq f',P(f,f')>0} F(f,f')\\ &= \min_{f\neq f',P(f,f')>0} p_f^* P(f,f')\\ &= \min_{f\neq f',P(f,f')>0} p_f^*\cdot\frac{q_{f,f'}}{\theta}\\ &\geq \frac{p_{\min}}{\theta}\cdot\alpha\exp(-\beta U_{\max}).\end{aligned} \tag{119}$$

Combining (116), (110), (111), and (119), we have

$$\begin{aligned}t_{\rm mix}(\epsilon) &\leq \frac{2}{\theta\Phi^2}[\ln\frac{1}{2\epsilon}+\frac{1}{2}\ln\frac{1}{p_{\min}}]\\ &\leq \frac{2\theta\exp(2\beta U_{\max})}{p_{\min}^2\alpha^2}[\ln\frac{1}{2\epsilon}+\frac{1}{2}\ln\frac{1}{p_{\min}}]\\ &\leq \frac{2}{\alpha}(M-1)NM^{2N}\exp(\beta(4U_{\max}-3U_{\min}))\\ &\quad[\ln\frac{1}{2\epsilon}+\frac{1}{2}N\ln M+\frac{1}{2}\beta(U_{\max}-U_{\min})].\end{aligned}$$

This concludes the proof for part (a).

(b) We present the proof for the upper bound of mixing time (57), which is based on the path coupling method [26].

Denote the set of APs as V and we have $|V| = N$. First, we obtain a discrete-time Markov chain by uniformization of continuous-time channel-hopping Markov chain. Denote this discrete-time Markov chain as $\mathcal{M}$. $\mathcal{M}$ is designed to sample from a given probability distribution $\boldsymbol{p^*}$ (43) on a state space $\mathcal{F}$. At each step, it selects an AP $v \in V$ uniformly at random and modified channel assignment for AP v. More precisely, when Markov chain $\mathcal{M}$ resides in a feasible configuration $f \in \mathcal{F}$, it does the following:

1) pick an AP $w \in V$ uniformly at random (with probability $\frac{1}{N} = \frac{1}{|V|}$), its assigned channel is denoted as f_w;
2) pick a new channel $c \in C - \{f_w\}$ uniformly at random (with probability $\frac{1}{M-1}$);
3) with probability $\frac{\exp(-\beta x_f)}{\sum_{f'\in\mathcal{F}}\exp(-\beta x_{f'})}$, let AP w switch to channel c. Otherwise, AP w sticks to channel f_w.

It can be shown that $\mathcal{M}$ has a transition matrix $P' = I + Q/\theta'$, where I is the identity matrix and

$$\theta' = \alpha N(M-1)\sum_{f\in\mathcal{F}}\exp(-\beta x_f). \tag{120}$$

It is not hard to see $\mathcal{M}$ is indeed a uniformized version of channel-hopping Markov chain.

Now we apply coupling method to bound the mixing time of $\mathcal{M}$. By a "coupling" for this chain, we mean a joint stochastic process (X_t, Y_t) on $\mathcal{F}\times\mathcal{F}$ such that each of the processes (X_t) and (Y_t) is a Markov chain on $\mathcal{F}$ with transition matrix P'. Typically, after defining the distance metric $d : \mathcal{F}\times\mathcal{F} \rightarrow \{0,1,\ldots,d_{\max}\}$, we try to construct a one-step distance-decreasing coupling $(X_0,Y_0)\rightarrow(X_1,Y_1)$ such that

$$E(d(X_1,Y_1)|X_0,Y_0) \leq \lambda\cdot d(X_0,Y_0) \tag{121}$$

for all $(X_0,Y_0)\in\mathcal{F}\times\mathcal{F}$, where $0 \leq \lambda < 1$. Applying this coupling iteratively results in a t-step coupling and a mixing time analysis.

In general, defining and analyzing a coupling for all pairs X_t, $Y_t \in \mathcal{F}$ is difficult. The path coupling technique [26] simplifies the approach by restricting attention to pairs in a connected subset $S \subseteq \mathcal{F}\times\mathcal{F}$. It then suffices to define a one-step coupling such that (121) holds for all $(X_0,Y_0)\in S$. Then the path coupling theorem [26] constructs, via simple compositions, a one-step coupling satisfying (121) for all $X_0, Y_0 \in \mathcal{F}$.

Given any two configurations $X, Y \in \mathcal{F}$, let $d(X,Y)$ denote the Hamming distance between X and Y, which equals to the number of APs at which the assigned channels are different. Now we denote S as configuration pairs $X, Y \in \mathcal{F}$ such that the channel assignments differ at exactly one AP. Then, we have

$$S = \{(X,Y)\in\mathcal{F}\times\mathcal{F} : d(X,Y) = 1\}.$$

For any AP $v \in V$, denote v^X as the channel assignment under configuration X. For example, if AP v chooses channel c under X, then $v^X = c$. Now we design a one-step coupling.

More precisely, consider a configuration pair $(X_0,Y_0)\in S$. Without loss of generality, we have

$$X_0 = (v_1^{X_0},\ldots,v_N^{X_0}),\quad Y_0 = (v_1^{Y_0},\ldots,v_N^{Y_0}),$$

where $v_j^{X_0} = v_j^{Y_0} \ \forall j = 2,\ldots,N$, and

$$v_1^{X_0} = a \in C,\quad v_1^{Y_0} = b \in C - \{a\}.$$

An AP $w \in V$ is chosen uniformly at random. At every step, both chains update the same AP w. Since there are M channels denoted by $C = \{c_1, c_2, \ldots, c_M\}$. Without loss of generality, we assume $a = c_1$ and $b = c_2$. Let $w^{X_0}(+)$ $\left(w^{Y_0}(+)\right)$ denote the channel that AP w selects and switches to under $X_0(Y_0)$. Here $w^{X_0}(+) = w^{X_0}(w^{Y_0}(+) = w^{Y_0})$ means AP w does not switch the channel under $X_0(Y_0)$.

The coupling for the update at time 1 is a coupling $(X_0,Y_0) \rightarrow (X_1,Y_1)$, where $(X_0,Y_0) \rightarrow (X_1,Y_1)$ denotes the channel switching operation. Then, the coupling is shown as follows.

1) If $w \neq v_1$, without loss of generality, we assume $w = v_n$, $2 \leq n \leq N$ and we have $w^{X_0} = w^{Y_0} = c_m$, $1 \leq$

$m \leq M$. We denote $p_{k,n} = Pr(w^{X_0}(+) = c_k)$ and $q_{k,n} = Pr(w^{Y_0}(+) = c_k), 1 \leq k \leq M$. Then, we have

$$p_{m,n} = Pr(w^{X_0}(+) = c_m) = 1 - \frac{\exp(-\beta x_{f_{X_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)}$$

for all $1 \leq k \leq M, k \neq m$,

$$p_{k,n} = Pr(w^{X_0}(+) = c_k) = \frac{1}{M-1} \cdot \frac{\exp(-\beta x_{f_{X_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)},$$

$$q_{m,n} = Pr(w^{Y_0}(+) = c_m) = 1 - \frac{\exp(-\beta x_{f_{Y_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)},$$

and for all $1 \leq k \leq M, k \neq m$,

$$q_{k,n} = Pr(w^{Y_0}(+) = c_k) = \frac{1}{M-1} \cdot \frac{\exp(-\beta x_{f_{Y_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)}.$$

We then take $r_{k,n} = \min\{p_{k,n}, q_{k,n}\}$ for any $k \in \{1, \ldots, M\}$. Now we define a random variable H_n satisfying

$$Pr(H_n = k) = \begin{cases} r_{k,n}, & \text{if } 1 \leq k \leq M; \\ 1 - \sum_{j=1}^{M} r_{j,n}, & \text{if } k = M+1; \\ 0, & \text{otherwise.} \end{cases}$$

We update X_0, Y_0 according to the following rules.

1) If $H_n = k$ where $1 \leq k \leq M$, then $w^{X_0}(+) = w^{Y_0}(+) = c_k$.
2) If $H_n = M+1$, then update X_0, Y_0 independently.
   a) $Pr(w^{X_0}(+) = c_k | H_n = M+1) = \frac{p_{k,n} - r_{k,n}}{1 - \sum_{j=1}^{M} r_{j,n}}$ $\forall k \in \{1, \ldots, M\}$.
   b) $Pr(w^{Y_0}(+) = c_k | H_n = M+1) = \frac{q_{k,n} - r_{k,n}}{1 - \sum_{j=1}^{M} r_{j,n}}$ $\forall k \in \{1, \ldots, M\}$.

2) Otherwise, $w = v_1$. We have

$$w^{X_0} = a \in C,$$
$$w^{Y_0} = b \in C - \{a\}.$$

Note that here $w^{X_0}(+) = a \ \left(w^{Y_0}(+) = b\right)$ means AP w does not switch the channel under $X_0 (Y_0)$. We denote $p_{k,1} = Pr(w^{X_0}(+) = c_k)$ and $q_{k,1} = Pr(w^{Y_0}(+) = c_k), 1 \leq k \leq M$. We have

$$p_{1,1} = Pr(w^{X_0}(+) = c_1 = a) = 1 - \frac{\exp(-\beta x_{f_{X_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)}$$

for all $2 \leq k \leq M$,

$$p_{k,1} = Pr(w^{X_0}(+) = c_k) = \frac{1}{M-1} \cdot \frac{\exp(-\beta x_{f_{X_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)},$$

$$q_{2,1} = Pr(w^{Y_0}(+) = c_2 = b) = 1 - \frac{\exp(-\beta x_{f_{Y_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)},$$

and for all $k \in \{1, 3, 4, \ldots, M\}$,

$$q_{k,1} = Pr(w^{Y_0}(+) = c_k) = \frac{1}{M-1} \cdot \frac{\exp(-\beta x_{f_{Y_0}})}{\sum_{f' \in \mathcal{F}} \exp(-\beta x_f)}.$$

We then take $r_{k,1} = \min\{p_{k,1}, q_{k,1}\}$ for any $k \in \{1, \ldots, M\}$. Now we define a random variable H_1 satisfying

$$Pr(H_1 = k) = \begin{cases} r_{k,1}, & \text{if } 1 \leq k \leq M; \\ 1 - \sum_{j=1}^{M} r_{j,1}, & \text{if } k = M+1; \\ 0, & \text{otherwise.} \end{cases}$$

We update X_0, Y_0 according to the following rules.

1) If $H_1 = k$ where $1 \leq k \leq M$, then $w^{X_0}(+) = w^{Y_0}(+) = c_k$.
2) If $H_1 = M+1$, then update X_0, Y_0 independently.
   a) $Pr(w^{X_0}(+) = c_k | H_1 = M+1) = \frac{p_{k,1} - r_{k,1}}{1 - \sum_{j=1}^{M} r_{j,1}}$, $\forall k \in \{1, \ldots, M\}$.
   b) $Pr(w^{Y_0}(+) = c_k | H_1 = M+1) = \frac{q_{k,1} - r_{k,1}}{1 - \sum_{j=1}^{M} r_{j,1}}$, $\forall k \in \{1, \ldots, M\}$.

It is not hard to show the one-step coupling designed above is a valid coupling. Now we analyze distance metric $E(d(X_1, Y_1) - 1 | X_0, Y_0) = 0$.

By the above coupling, we know that when $w = v_n, 1 \leq n \leq N$, $Pr(w^{X_0}(+) = w^{Y_0}(+)) = \sum_{j=1}^{M} r_{j,n}$. Then, we have

$$E[d(X_1, Y_1) - 1 | X_0, Y_0, w = v_1] = -\sum_{j=1}^{M} r_{j,1},$$

$$E[d(X_1, Y_1) - 1 | X_0, Y_0, w = v_n] = 1 - \sum_{j=1}^{M} r_{j,n}, \ \forall 2 \leq n \leq N.$$

It follows that

$$\begin{aligned} &E[d(X_1, Y_1) - 1 | X_0, Y_0] \\ &= \sum_{n=1}^{N} P(w = v_n) \cdot E[d(X_1, Y_1) - 1 | X_0, Y_0, w = v_n] \\ &= \frac{1}{N} \cdot \left[N - 1 - \sum_{n=1}^{N} \sum_{j=1}^{M} r_{j,n} \right]. \end{aligned}$$

On the other hand, we can see that

$$\begin{aligned} &\sum_{j=1}^{M} r_{j,1} \geq \frac{M}{M-1} \cdot \frac{1}{|\mathcal{F}|} \cdot \exp(-\beta(U_{\max} - U_{\min})), \\ &\sum_{j=1}^{M} r_{j,n} \geq 1 - \frac{\exp(\beta(U_{\max} - U_{\min}))}{|\mathcal{F}|} \\ &\qquad + \frac{\exp(-\beta(U_{\max} - U_{\min}))}{|\mathcal{F}|}, \forall 2 \leq n \leq N. \end{aligned}$$

Therefore,

$$\begin{aligned} &E[d(X_1, Y_1) - 1 | X_0, Y_0] \\ &= \frac{1}{N} \cdot \left[N - 1 - \sum_{n=1}^{N} \sum_{j=1}^{M} r_{j,n} \right] \\ &\leq \frac{1}{N} \cdot \frac{1}{|\mathcal{F}|} \exp(-\beta(U_{\max} - U_{\min})) \\ &\quad \cdot [(N-1)\exp(2\beta(U_{\max} - U_{\min})) - (N + \frac{1}{M-1})]. \end{aligned}$$

For convenience, let

$$K = \frac{[(N+\frac{1}{M-1}) - (N-1)\exp(2\beta(U_{\max}-U_{\min}))]}{\exp(\beta(U_{\max}-U_{\min}))\cdot|\mathcal{F}|}, \tag{122}$$

which is positive when

$$0 < \beta < \frac{1}{2(U_{\max}-U_{\min})}\ln(\frac{N+\frac{1}{M-1}}{N-1}).$$

Then, it follows that for any $(X_0, Y_0) \in S$

$$\begin{aligned} E[d(X_1,Y_1)|X_0,Y_0] &< 1-\frac{K}{N} \\ &= (1-\frac{K}{N})\cdot d(X_0,Y_0). \end{aligned}$$

By path coupling theorem [26], we know that for any $(X_0, Y_0) \in \mathcal{F}\times\mathcal{F}$,

$$\begin{aligned} E[d(X_1,Y_1)|X_0,Y_0] &< (1-\frac{K}{N})\cdot d(X_0,Y_0) \\ &= \lambda \cdot d(X_0,Y_0) \end{aligned}$$

where $\lambda = 1-\frac{K}{N}$.

Applying this one-step coupling iteratively results in a t-step coupling, and we have for any t, $(X_t, Y_t) \in \mathcal{F}\times\mathcal{F}$,

$$\begin{aligned} P[X_t \neq Y_t] &= P[d(X_t,Y_t) \geq 1] \\ &\leq E[d(X_t,Y_t)] \\ &\leq \lambda^t \cdot \mathrm{diam}(\mathcal{F}) \\ &\leq N\cdot\lambda^t. \end{aligned}$$

Thus, for discrete-time Markov chain $\mathcal{M}$,

$$d_{TV}(P^t(X_0,\cdot), P^t(Y_0,\cdot)) \leq N\cdot\lambda^t.$$

Then by (120), (122), and uniformization theorem [16], we know that for any $f \in \mathcal{F}$,

$$\begin{aligned} &d_{TV}(\boldsymbol{H}_t(f), \boldsymbol{p}^*) \\ &= d_{TV}[\sum_{j=0}^{\infty}\frac{(\theta' t)^j}{j!}\exp(-\theta' t)P^j(f,\cdot), \boldsymbol{p}^*] \\ &\leq \sum_{j=0}^{\infty}\frac{(\theta' t)^j}{j!}\exp(-\theta' t)d_{TV}(P^j(f,\cdot), \boldsymbol{p}^*) \\ &\leq N\cdot\sum_{j=0}^{\infty}\frac{(\theta' t\lambda)^j}{j!}\exp(-\theta' t) \\ &= N\cdot\exp(-\theta'(1-\lambda)t) \\ &= N\cdot\exp\left(-\theta'\cdot\frac{Kt}{N}\right) \\ &\leq N\cdot\exp\left(-\alpha K'(M-1)t\exp\left(-\beta\left(2U_{\max}-U_{\min}\right)\right)\right), \end{aligned}$$

where $k' = N + \frac{1}{M-1} - (N-1)\exp(2\beta(U_{\max}-U_{\min})$.

Thus, we have

$$t_{\mathrm{mix}}(\epsilon) \leq \frac{\frac{1}{\alpha(M-1)}\cdot\exp\left(\beta(2U_{\max}-U_{\min})\right)\cdot\ln\frac{N}{\epsilon}}{N+\frac{1}{M-1}-(N-1)\exp(2\beta(U_{\max}-U_{\min})}.$$

This concludes the proof for part (b).

References

[1] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[2] S. Liu, M. Chen, S. Sengupta, M. Chiang, J. Li, and P. Chou, "P2P streaming capacity under node degree bound," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst.*, 2010, pp. 587–598.

[3] S. Liew, C. Kai, H. Leung, and P. Wong, "Back-of-the-envelope computation of throughput distributions in CSMA wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 9, pp. 1319–1331, Sep. 2010.

[4] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2005, pp. 23–34.

[5] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.*, 2008, pp. 1511–1519.

[6] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. Poor, "Towards utility-optimal random access without message passing," *Wireless Commun. Mobile Comput.*, vol. 10, no. 1, pp. 115–128, 2010.

[7] S. Rajagopalan and D. Shah, "Distributed algorithm and reversible network," in *Proc. 42nd Annu. Conf. Inf. Sci. Syst.*, 2008, pp. 498–502.

[8] J. Walrand, "Entropy in communication and chemical systems," in *Proc. First Int. Symp. Applied Sci. Biomed. Commun. Technol.*, 2008, pp. 1–5.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. press, 2004.

[10] P. Laarhoven and E. Aarts, *Simulated Annealing: Theory and Applications*. New York, NY: Springer-Verlag, 1987.

[11] S. Chau, M. Chen, and S. Liew, "Capacity of large scale CSMA wireless networks," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 97–108.

[12] S. Liu, T. Basar, and R. Srikant, "Controlling the Internet: A survey and some new results," presented at the IEEE Conf. Decision Control, Maui, HI, 2003.

[13] L. Jiang and J. Walrand, Convergence and stability of a distributed CSMA algorithm for maximal network throughput Univ. of California, Berkeley, CA, Tech. Rep. UCB/EECS-2009-43, Mar. 2009 [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-43.html

[14] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.

[15] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY, USA: Springer-Verlag, 2003.

[16] M. Kijima, *Markov Processes for Stochastic Modeling*. Boca Raton, FL: CRC Press, 1997.

[17] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of Markov chains," *Annu. Appl. Probability*, vol. 1, pp. 36–61, 1991.

[18] R. Srikant, *The Mathematics of Internet Congestion Control*. Cambridge, MA: Birkhauser, 2004.

[19] P. Key, L. Massoulie, and P. Towsley, "Path selection and multipath congestion control," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2007, pp. 143–151.

[20] J. Li, P. Chou, and C. Zhang, "Mutualcast: An efficient mechanism for one-to-many content distribution," presented at the ACM SIGCOMM Asia Workshop, 2005, Beijing, China.

[21] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. Chou, "Peer-to-peer streaming capacity," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5072–5087, Aug. 2011.

[22] Y. Wang, T. Fu, and D. Chiu, "Design and evaluation of load balancing algorithms in P2P streaming protocols," *Comput. Netw.*, vol. 55, pp. 4043–4054, 2011.

[23] R. Xia and J. Muppala, "A survey of BitTorrent performance," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 140–158, Apr.–Jun 2010.

[24] S. Zhang, Z. Shao, and M. Chen, "Optimal distributed P2P streaming under node degree bounds," in *Proc. IEEE Int. Conf. Netw. Protocol*, 2010, pp. 253–262.

[25] D. Levin, Y. Peres, and E. Wilmer, *Markov Chains and Mixing Times*. Providence, RI: American Mathematical Society, 2009.

[26] R. Bubley and M. Dyer, "Path coupling: A technique for proving rapid mixing in Markov chains," in *Proc. 38th Annu. Symp. Found. Comput. Sci.*, 1997, pp. 223–231.

[27] Z. Shao, M. Chen, A. S. Avestimehr, and S. Li, "Cross-layer optimization for wireless networks with deterministic channel models," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5840–5862, Aug. 2011.

[28] Z. Shao, H. Zhang, M. Chen, and K. Ramchandran, "Reverse-engineering BitTorrent: A Markov approximation perspective," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2996–3000.

[29] H. Zhang, Z. Shao, M. Chen, and K. Ramchandran, "Optimal neighbor selection in BitTorrent-like peer-to-peer networks," in *Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Modeling Comput. Syst.*, 2011, pp. 141–142.
[30] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint VM placement and routing for data center traffic engineering," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2876–2880.
[31] Z. Shao, X. Jin, W. Jiang, M. Chen, and M. Chiang, "Intra-data-center traffic engineering with ensemble routing," presented at the IEEE Int. Conf. Computer Communications, Torino, Italy, 2013.
[32] F. Kelly*, Reversibility and Stochastic Networks*. Chichester, U.K.: Wiley, 1979.
[33] D. Bertsekas and J. Tsitsiklis*, Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

Minghua Chen (SM'12) received his B.Eng. and M.S. degrees from the Department of EE at Tsinghua University in 1999 and 2001, respectively. He received his Ph.D. degree from the Department of EECS at UC Berkeley in 2006. He spent one year visiting Microsoft Research Redmond as a Postdoc Researcher. He joined the Department of Information Engineering, the Chinese University of Hong Kong in 2007 as an Assistant Professor. He is also an Adjunct Associate Professor in Peking University Shenzhen Graduate School in 2011–2014. He received the Eli Jury Doctoral Thesis award from UC Berkeley in 2007, the ICME Best Paper Award in 2009, the IEEE TRANSACTIONS ON MULTIMEDIA Best Paper Award in 2009, and the ACM Multimedia Best Paper Award in 2012. His current research interests include smart (micro) grids, energy-efficient data centers, distributed and stochastic network optimization and control, multimedia networking, p2p networking, wireless networking, multi-level trust data privacy, network coding, and secure network communications.

Soung-Chang Liew (F'12) received his S.B., S.M., E.E., and Ph.D. degrees from the Massachusetts Institute of Technology. From 1984 to 1988, he was at the MIT Laboratory for Information and Decision Systems, where he investigated Fiber-Optic Communications Networks. From March 1988 to July 1993, he was at Bellcore (now Telcordia), NJ, where he engaged in Broadband Network Research. He has been a Professor at the Department of Information Engineering, the Chinese University of Hong Kong, since 1993. He is an Adjunct Professor at Peking University and Southeast University, China.

Prof. Liew's research interests include wireless networks, Internet protocols, multimedia communications, and packet switch design. Prof. Liew's research group won the best paper awards in IEEE MASS 2004 and IEEE WLN 2004. Separately, TCP Veno, a version of TCP to improve its performance over wireless networks proposed by Prof. Liew's research group, has been incorporated into a recent release of Linux OS. In addition, Prof. Liew initiated and built the first inter-university ATM network testbed in Hong Kong in 1993. More recently, Prof. Liew's research group pioneers the concept of Physical-layer Network Coding (PNC).

Besides academic activities, Prof. Liew is also active in the industry. He co-founded two technology start-ups in Internet Software and has been serving as a consultant to many companies and industrial organizations. He is currently consultant for the Hong Kong Applied Science and Technology Research Institute (ASTRI), providing technical advice as well as helping to formulate R& D directions and strategies in the areas of Wireless Internetworking, Applications, and Services.

Prof. Liew is the holder of eight U.S. patents and a Fellow of IEEE, IET and HKIE. He currently serves as Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and Ad Hoc and Sensor Wireless Networks. He is the recipient of the first Vice-Chancellor Exemplary Teaching Award at the Chinese University of Hong Kong. Publications of Prof. Liew can be found in wwwiecuhkeduhksoung.

Ziyu Zhao (S'09–M'11) received the B.S. and M.S. degrees in electrical engineering from Peking University, Beijing, China, in 2001 and 2004, respectively and the Ph.D. degree in information engineering from The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, in 2010. Currently, he is a Postdoctoral Fellow at The Chinese University of Hong Kong. His research interests include network coding, information theory, cloud computing, mobile computing, network optimization and distributed algorithm design.

Caihong Kai received her master's degree in Electronic Engineering and Computer Science from University of Science and Technology of China in 2006 and her Ph.D. in Information Engineering from The Chinese University of Hong Kong in 2010, respectively. Caihong is now an Associate Professor of the School of Computer and Information, Hefei University of Technology. Her research interests are in Wireless Communications and Networking.