Markov Approximation for Combinatorial Network Optimization

Minghua Chen, Soung Chang Liew, Ziyu Shao, and Caihong Kai
Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
Email: {minghua, soung, zyshao6, chkai6}@ie.cuhk.edu.hk

***Abstract*—Many important network design problems can be formulated as a combinatorial optimization problem. 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. Certain carefully designed Markov chains among this class yield distributed algorithms that solve the log-sum-exp approximated combinatorial network optimization problem. By three case studies, we illustrate that Markov approximation technique not only can provide fresh perspective to existing distributed solutions, but also can help us generate new distributed algorithms in various domains with provable performance. We believe the Markov approximation techniques will find application in many network optimization problems, and this paper serves as a call for participation of it.**

I. Introduction

Many important network design and resource allocation problems can be formulated as a combinatorial network optimization problem. Two well-studied examples are

- 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].
- The optimal path slection problem in traffic engineering of finding the "best" set of paths for every user to maximize the overall network throughput [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) the optimization problem could be amenable to centralized implementation only.

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. 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 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).

Historically, our investigation of the Markov-approximation technique was inspired by the recent progress in carrier-sense multiple-access (CSMA) network design. In [3] [4], it was shown that the throughput of links in a CSMA network can be computed from a time-reversible Markov chain. Refs. [5] [6] reverse-engineered to show that CSMA solves the combinatorial MWIS problem asymptotically, off by an entropy term. With this observation, Refs. [5] [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 to use the entropy term as a forward-engineering device to stimulate new algorithms for various network combinatorial problems. This expands the usefulness of the approach originally expounded in the series of work in [3]–[7], [9] to many other domains beyond CSMA networks. In particular, this paper makes the following contributions:

- it shows that an entropy term appears as a direct consequence of our approximating the optimal value of *any* combinatorial problem using a log-sum-exp function.
- 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).
- it shows that certain carefully designed time-reversible Markov chains among this class yield distributed algorithms that solve the log-sum-exp approximated problem.
- it demonstrates the usage of the Markov approximation technique by considering two specific problems 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 to Wireless LANs located in the vicinity of each other.

This work was supported General Research Funds (Project Number 411008, 411209, and 414507), Area of Excellence grant on Institute of Network Coding, established under the University Grant Committee of the Hong Kong Administative Region, China, Direct Grants (Project Number 2050397 and 2050436) from the Chinese University of Hong Kong, and and two gift grants from Microsoft and Cisco.

The rest of this paper is organized as follows. We first present the Markov approximation 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] [6]. The goal of Section III is to provide a new perspective to the design of an existing distribution solution. In Sections IV and V, our goal shift to that of applying Markov approximation to synthesize new distributed algorithms in problem domains. Sections IV studies the optimal path selection problem in multipath transmission over wireline networks. Sections 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 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 following combinatorial optimization problem[2]

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

An equivalent formulation is

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

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

Many practical and important problems, or their subproblems, can be formulated into 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, IV, and V.

For many problems, formulation in (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, but in a distributed manner. Systems running distributed algorithms are more robust to user and system dynamics than those running centralized algorithms.

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

[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 problem. In this paper we focus on the standard form given in (1).

B. Log-sum-exp Approximation

To gain insights on the structure of the problem **MWC**, we approximate the max function in (1) by a differentiable function as follows:

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

where β is a positive constant and $\boldsymbol{x} \triangleq [\sum_{r\in R} x_i(f), f \in \mathcal{F}]$. This approximation is known as the convex log-sum-exp approximation to the max function . Its accuracy is known as follows.

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

$$\begin{aligned} \max(y_1,\ldots,y_n) &\leq \frac{1}{\beta}\log\left(\exp(\beta y_1)+\cdots+\exp(\beta y_n)\right) \\ &\leq \max(y_1,\ldots,y_n) + \frac{1}{\beta}\log n. \end{aligned} \tag{4}$$

Hence, $\max(y_1,\ldots,y_n) = \lim_{\beta\to\infty} \frac{1}{\beta}\log\left(\exp(\beta y_1)+\cdots+\exp(\beta y_n)\right).$

Proof: Since for any $\beta > 0$,

$$\begin{aligned} \exp\left(\max(\beta y_1,\ldots,\beta y_n)\right) &\leq \exp(\beta y_1) + \ldots + \exp(\beta y_n) \\ &\leq n \exp\left(\max(\beta y_1,\ldots,\beta y_n)\right). \end{aligned} \tag{5}$$

We have

$$\begin{aligned} \max(\beta y_1,\ldots,\beta y_n) &\leq \log\left(\exp(\beta y_1) + \ldots + \exp(\beta y_n)\right) \\ &\leq \max(\beta y_1,\ldots,\beta y_n) + \log(n) \end{aligned} \tag{6}$$

By dividing β in both sides, we obtain inequality (4). When $\beta \to \infty$, we get the desired equality. ∎

We summarize some important observations of $g_\beta(\boldsymbol{x})$ in the following theorem. Some of these observations were also found relevant in the context of Geometric Programming [10].

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

- its conjugate function[3] is given by

$$g^*_\beta(\boldsymbol{p}) = \begin{cases} \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f & \text{if } \boldsymbol{p} \geq 0 \text{ and } \mathbf{1}^T\boldsymbol{p} = 1 \\ \infty & \text{otherwise.} \end{cases} \tag{7}$$

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

$$\begin{aligned} g_\beta(\boldsymbol{x}) = \quad & \max_{\boldsymbol{p}\geq 0} \sum_{f\in\mathcal{F}} p_f \sum_{r\in R} x_r(f) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \\ & \text{s.t.} \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned} \tag{8}$$

Proof: The proof of (7) follows nearly the same way as [11, pp.93]. $g_\beta(\boldsymbol{x})$ is a convex function because the log-sum-exp function is a convex function [11]. 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 [11, Section 3.3.2], the conjugate of its conjugate $g^*_\beta(\boldsymbol{p})$ is itself, i.e., (8) holds. ∎

[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 $g^*(\boldsymbol{z}) = \sup_{\boldsymbol{y}\in\mathbf{dom}g}\left(\boldsymbol{z}^T\boldsymbol{y} - g(\boldsymbol{y})\right)$ [11].

Remark: Several observations can be made. First, by the log-sum-exp approximation in (3), we are implicitly solving an approximated version of the problem **MWC – EQ**, off by an *entropy* term $-\frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f$[4]. The optimality gap is thus bounded by $\frac{1}{\beta}\log|\mathcal{F}|$, where $|\mathcal{F}|$ represents the size of $\mathcal{F}$. We emphasize that this is a direct consequence of we theoretically approximating the max function by a log-sum-exp function in (3). 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 case studies later, usually there are practical constraints or overhead concerns of using large β. Third, we can derive a close-form of the optimal solution of the problem in (8). Let λ be the Lagrange multiplier associated with the equality constraint in (8) and $p^*_f(\boldsymbol{x})$, $f \in \mathcal{F}$ be the optimal solution of the problem in (8). By solving the Karush-Kuhn-Tucker (KKT) conditions [11] of the problem in (8):

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

$$\sum_{f\in\mathcal{F}} p^*_f(\boldsymbol{x}) = 1, \tag{10}$$

$$\lambda \geq 0, \tag{11}$$

we have

$$p^*_f(\boldsymbol{x}) = \frac{\exp\left(\beta\sum_{r\in R} x_r(f)\right)}{\sum_{f'\in\mathcal{F}}\exp\left(\beta\sum_{r\in R} x_r(f')\right)}, \forall f \in \mathcal{F}. \tag{12}$$

By time-sharing among different configurations f according to their portions $p^*_f(\boldsymbol{x})$, we solve the problem **MWC – EQ**, and hence the problem **MWC**, approximately. We remark the optimality gap is bounded by $\frac{1}{\beta}\log|\mathcal{F}|$, which can be made small by choosing large β.

C. Algorithm Design via Markov Chain

A key to create new algorithm designs is to treat $p^*_f(\boldsymbol{x})$ $(f \in \mathcal{F})$ as the stationary distribution of a time-reversible Markov chain. Time-reversible Markov chains usually have structures that allows distributed implementation. As the Markov chain converges to its stationary distribution, we approach $p^*_f(\boldsymbol{x})$ in a distributed manner.

Lemma 1: For any probability distribution of the product form $p^*_f(\boldsymbol{x})$ in (12), 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 by the product form $p^*_f(\boldsymbol{x})$ in (12).

The proof is relegated to Appendix-A.

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

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

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

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-pair between two states, are also time-reversible. Furthermore, all Markov chains share the same stationary distribution.

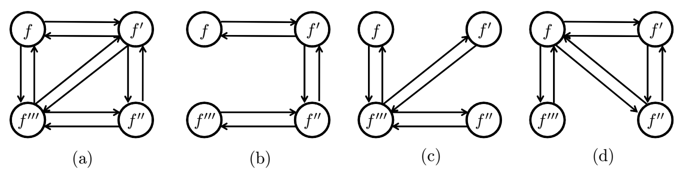


Fig. 1. The Markov chains in (b), (c), (d), by adding/removing transition edge-pair between two states in the time-reversible Markov chain in (a), are also time-reversible. All Markov chains have the same stationary distribution.

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

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

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

$q_{f',f}$ is defined in a symmetric way.

OPT2: let $q_{f,f'}$ be positive correlated to the system performance under the targeting configuration f', specifically,

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

$q_{f',f}$ is defined in a symmetric way.

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

$$q_{f,f'} = \alpha\exp\left(\frac{1}{2}\beta\sum_{r\in R}\left(x_r(f') - x_r(f)\right)\right). \tag{16}$$

$q_{f',f}$ is defined in a symmetric way.

[4]Under the context of CSMA scheduling, Jiang and Walrand [5] arrive a similar observation using a different approach. We will discuss more details when we come to CSMA utility maximization in Section III.

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

$$q_{f,f'} = \alpha \exp\left(\beta \sum_{r\in R} (x_r(f') - x_r(f))\right). \quad (17)$$

Design option OPT1 implies the transition rate from f to f', i.e., $q_{f,f'}$, is *independent* of the performance under targeting configuration f'. In contrast, $q_{f,f'}$ in OPT2 depends only on the performance of targeting 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 targeting configuration f' a prior, 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 rate fitting into option OPT4.

Recalled 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 clock and wait for a random amount of time before they switching local configurations, we can design transition to happen only between two configurations f and f' that differ by one user's local configuration. If individual users can collect system performance that its switching can affect in a distributed manner, then the Markov chain can be implemented in a distributed manner.

Note Simulated Annealing [12] also uses Markov chain 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 algorithm to solve the optimization problem approximately. In the following sections, we study three cases to illustrate how such design are done.

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] [6]. By doing so, we wish to provide new perspective to the design of existing distributed solutions.

A. Settings

Consider a hidden-node-free [5]and collision-free 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 the *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.

[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 mechanism, 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 its throughput analysis is more tractable. As studied in [13], 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.

B. Wireless Utility Maximization Problem

Consider the following utility maximization problem over G_c. Note here no wireless protocol is assumed.

$$\begin{aligned} \textbf{MP}: \quad & \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \sum_{s\in S} U_s(z_s) && (18)\\ \text{s.t.} \quad & \sum_{s:l\in s, s\in S} z_s \leq \sum_{f:l\in f} p_f, \quad \forall l \in L \\ & \sum_{f\in\mathcal{F}} p_f = 1 \end{aligned}$$

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 aggregate incoming rate of every link can not exceed the average link throughput. By relaxing the first set of inequality constraints, we get its partial Lagrangian as follows

$$L(\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{\lambda}) = \sum_{s\in S} U_s(z_s) - \sum_{l\in L} \lambda_l \left(\sum_{s:l\in s, s\in S} z_s - \sum_{f:l\in f} p_f \right), \quad (19)$$

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 a concave optimization one and the 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 $\boldsymbol{\lambda}$:

$$\begin{aligned} \min_{\boldsymbol{\lambda}\geq 0} \max_{\substack{\boldsymbol{z}\geq 0\\ \boldsymbol{p}\geq 0}} \quad & \sum_{s\in S} U_s(z_s) - \sum_{l\in L} \lambda_l \sum_{\substack{s:l\in s,\\ s\in S}} z_s + \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l && (20)\\ \text{s.t.} \quad & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned}$$

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

$$\begin{aligned} \textbf{MWIS}: \quad & \max_{\boldsymbol{p}\geq 0} \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l && (21)\\ \text{s.t.} \quad & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned}$$

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

C. Approach by Markov Approximation

Observing the problem **MWIS** is a combinatorial optimization problem, we apply the Markov Approximation. First, we apply the log-sum-exp approximation

$$\max_{f\in\mathcal{F}} \sum_{l\in f} \lambda_l \approx \frac{1}{\beta} \log\left[\sum_{f\in\mathcal{F}} \exp\left(\beta \sum_{l\in f} \lambda_l\right)\right], \quad (22)$$

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

$$\begin{aligned} \max_{\boldsymbol{p}\geq 0} \quad & \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l - \frac{1}{\beta} \sum_{f\in\mathcal{F}} p_f \log p_f \quad (23) \\ \text{s.t.} \quad & \sum_{f\in\mathcal{F}} p_f = 1, \end{aligned}$$

and the corresponding (unique) optimal solution is

$$p_f(\boldsymbol{\lambda}) = \frac{\exp\left(\beta \sum_{l\in f} \lambda_l\right)}{\sum_{f'\in\mathcal{F}} \exp\left(\beta \sum_{l\in f'} \lambda_l\right)}, \forall f \in \mathcal{F}. \quad (24)$$

We first study the impact of solving the subproblem **MWIS** approximately by (22).

1) Entropy Term as A Consequence of Log-sum-exp Approximation: It is unlikely that we are still solving the original problem **MP**. After we approximate problem **MWIS** by the problem in (23), the partial Lagrangian problem in (20) turns into

$$\begin{aligned} \min_{\boldsymbol{\lambda}\geq 0} \quad \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \quad & \sum_{s\in S} U_s(z_s) - \frac{1}{\beta} \sum_{f\in\mathcal{F}} p_f \log p_f \quad (25) \\ & - \sum_{l\in L} \lambda_l \left(\sum_{s: l\in s, s\in R} z_s - \sum_{f: l\in f} p_f \right) \\ \text{s.t.} \quad & \sum_{f\in\mathcal{F}} p_f = 1. \quad (26) \end{aligned}$$

It can be verified to be the partial Lagrangian problem of the following primal problem:

$$\begin{aligned} \mathbf{MP-MA}: \quad \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \quad & \sum_{s\in S} U_s(z_s) - \frac{1}{\beta} \sum_{f\in\mathcal{F}} p_f \log p_f \quad (27) \\ \text{s.t.} \quad & \sum_{f: l\in f} p_f \geq \sum_{s: l\in s, s\in R} z_s, \quad \forall l \in L \\ & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned}$$

Comparing problems **MP – MA** and **MP**, we observe that when we approximate the subproblem **MWIS** by the one in (23), we are in effect approximating the problem **MP** by problem **MP – 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 (22), independent of any wireless protocol, e.g., CSMA, to be used.

Historically, by modeling and studying the carrier sensing behavior, the authors of [3], [4] 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 (24). The authors of [5], [6] then reverse-engineered $p_f(\boldsymbol{\lambda})$ in (24) to be the optimal solution to the problem in (23). With this observation, the authors of [5], [6] design distributed algorithms on top of CSMA to solve problem **MP – MA**, an entropy term away from problem **MP**.

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 now design a time-reversible Markov chain whose stationary distribution is given by (24) and work out its distributed implementation.

The states of the Markov chain are the independent sets over G_c. To make sure the network operates over only the independent sets, any two interfered links (in particular their transmitters) must be able to sense each other so one will keep silence while the other is transmitting. This can be done distributedly by each transmitter sensing its receiving power and only starting its transmission if the power is below a properly selected threshold [13].

We 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,

a) we set $q_{f,f'}$ to zero, if one of f or f' is not a subset of the other (i.e., $|f|-|f'| = \pm 1$ is not satisfied). 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 $f' = f \cup \{l'\}$,

b) we set $q_{f',f}$ to 1, and

$$q_{f,f'} = \exp\left(\beta\left(\sum_{l\in f'} \lambda_l - \sum_{l\in f} \lambda_l\right)\right) = \exp\left(\beta\lambda_{l'}\right). \quad (28)$$

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, link l' will freeze its countdown process. When the transmission is over, link l' countdown according to the residual back-off time, which is still exponential distributed with the same rate $\exp(\beta\lambda_{l'})$ because of the memoryless property of exponential distribution.

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] [6] and in Section III-C4.

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. This thought exercise raises out a significant point. Namely, had the CSMA protocol not been invented previously, the Markov Approximation technique might have led us to it, starting with the premise that we wanted to find an approximate distributed algorithm to problem **MP**. A similar exercise on other problem domains in which a satisfactory distributed solution is still lacking may help us to discover new distributed algorithms. That is, the Markov approximation technique is a general framework.

3) Approximation Accuracy Limited by Physical Constraints: Mathematically, as β approaches infinity, we should be able to solve **MWIS** exactly. However, there are certain physical constraints preventing β to be too large. In CSMA networks, The value of $\exp(\beta\lambda_{l'})$ corresponds to the ratio of average packet duration to average backoff time [3]. For a given fixed packet duration (e.g., that corresponds to the maximum size of an Internet packet), increasing β basically means decreasing the average back off time. However, the average backoff time cannot be arbitrarily decreased. In practical situation, the backoff process is actually time-slotted. Each backoff time slot σ must be sufficient large either due to 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 progagation delay) for CSMA to operate properly. If we assume the radius of a WLAN coverage is 75m and the speed of light $c = 3 \times 10^8$ m/s, then $\sigma \geq 1\mu s$. In 802.11b, $\sigma = 20\mu s$, but here we will assume the fundamental limit of $\sigma = 1\mu s$.

Typically, we should allow for enough 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 $15 \times 1\mu s = 15\mu s$.

Now suppose that the data rate of the WLAN is 10 Mbps, and the average packet size is 1kB. The packet duration is then in the ballpark of $800\mu s$ (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 10 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 **MP – MA** *by CSMA and Primal-dual Algorithm:* With the approximated optimal value to problem **MWIS** in Equation , we can solve the following problem to get the optimal solution $\boldsymbol{z}^*$ and $\boldsymbol{\lambda}^*$ (and thus $\boldsymbol{p}^*$):

$$\sum_{s\in S} U_s(z_s) - \sum_{l\in L} \lambda_l \sum_{s:l\in s, s\in S} z_s + \frac{1}{\beta} \log\left[\sum_{f\in\mathcal{F}} \exp\left(\beta \sum_{l\in f} \lambda_l\right)\right]. \quad (29)$$

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], [6]. We study a primal-dual algorithm as follows

$$\begin{cases} \dot{\lambda}_l = k_l \left[\sum_{s:l\in s, s\in S} z_s - \sum_{l\in f} p_f(\beta\boldsymbol{\lambda})\right]^+_{\lambda_l} \\ \dot{z}_s = \alpha_s \left[U'_s(z_s) - \sum_{l\in s} \lambda_l\right]^+_{z_s} \end{cases}, \quad (30)$$

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 $\sum_{l\in s} p_s(\beta\boldsymbol{\lambda})$ is the stationary throughput of link l, by running CSMA protocol network-widely with transmission aggressive vector $\beta\boldsymbol{\lambda}$. This is a key observation made in [3] [5] [6]. The Lagrange dual variable λ_l can then be updated based on information of the local queue at link l.

Given the Markov chain converges to its stationary distribution instantaneously, proving the convergence of the algorithm in (30) can be done by a standard technique using Lyapunov function [14],

In practice, however, the Markov chain may not converge before the primal-dual algorithm (30) evolves. the algorithm then turns into a stochastic primal-dual algorithm, given as follows:

$$\lambda_l(m+1) = \left[\lambda_l(m) + \epsilon(m)\left(\sum_{s:l\in s, s\in S} z_s(m) - \overline{\theta}_l(m)\right)\right]_+ \quad \forall l \in L,$$
$$z_s(m+1) = \left[z_s(m) + \epsilon(m)\left(U'_s(z_s(m)) - \sum_{l:l\in s} \lambda_l(m)\right)\right]_+ \quad \forall s \in S, \quad (31)$$

where $\epsilon(m)$ is the step size, $\overline{\theta}_l(m)$ is the average link rate measured by link l within the update interval T_m, and T_m is the time interval between the system updating $(\boldsymbol{\lambda}(m-1), \boldsymbol{z}(m-1))$ and $(\boldsymbol{\lambda}(m), \boldsymbol{z}(m))$. The primal-dual algorithm (30) can be considered a continuous time approximation of (31) with small $\epsilon(m)$ and T_m.

Under suitable choices of step sizes and update intervals, we establish the convergence of the stochastic primal-dual algorithm (31) with probability one in the following theorem.

Theorem 2: Assume that $U'_s(0) < \infty, \forall s \in S$, $\max_{s,m} z_s(m) < \infty$ and $\max_{l,m} \lambda_l(m) < \infty$. The stochastic primal-dual algorithm in (31) converges to the optimal solutions of **MP – MA** asymptotically with probability one under the following conditions on step sizes and update intervals:

$$\{T_m\} \text{ is non-decreasing with m}, \quad (32)$$
$$\epsilon(m) > 0\ \forall m,\ \sum_{m=1}^{\infty} \epsilon(m) = \infty,\ \sum_{m=1}^{\infty} \epsilon^2(m) < \infty, \quad (33)$$
$$\sum_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty. \quad (34)$$

Further, the setting $\epsilon(1) = T(1) = 1, \epsilon(m) = \frac{1}{m},\ T_m = m,\ m \geq 2$ is one specific choice satisfying conditions (32)-(34).

The proof is relegated to Appendix-B. Inspired by and similar to [15], we also adopt the standard methods of stochastic approximation [16] and Markov chain [17], [18]. The difference between our proof and [15] is that, our proof studies the saddle points of Lagrangian function, while [15] studies the optimal dual solutions directly.

IV. Case 2: Path Selection in Wireline Networks

A. Settings

Consider a wireline network *G=(V, L)*, 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 a user s selects from J_s, it opens a connection to transfer data. Maintaining connections and paths consume users' resources and incur overhead. Due to the limited system resource or concern on overhead, each user $s \in S$ operates at most D_s connections over D_s paths.

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 to be $J_{s,f} \subseteq J_s$, where $|J_{s,f}| = D_s$. Similar to [2], we assume that there exists at most one bottleneck along each path, where "bottleneck" is defined as the link shared among multiple paths. Therefore, at most one link of a path will be shared with other paths. While a limited assumption that may not hold in practice, it is a reasonable model for some realistic scenarios [19]. Even under this assumption, path selection is still a challenging problem [2]. We also assume the utility functions to be twice differentiable, increasing and strictly concave.

B. Joint Path Selection and Multipath Utility Maximization

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 of the long-term throughputs:

$$\begin{aligned} \mathbf{PS}: \quad & \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \sum_{s\in S} U_s(z_s) \\ \text{s.t.} \quad & z_s \leq \sum_{f\in\mathcal{F}} R_{s,f}\, p_f \quad \forall s \in S \\ & \sum_{f\in\mathcal{F}} p_f = 1, \end{aligned} \tag{35}$$

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 named "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 [2]:

$$\begin{aligned} \mathbf{MP-UCC}: \quad & \max_{\boldsymbol{y}\geq \mathbf{0}} \sum_{s\in S} \sum_{j\in J_{s,f}} U_s(y_j) \\ \text{s.t.} \quad & \sum_{j:l\in j} y_j \leq C_l, \quad \forall l \in L_f \end{aligned} \tag{36}$$

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 **MP – UCC** denoted by $\hat{y}_j, j \in J_{s,f}, s \in S$, the equilibrium capacity is given by

$$R_{s,f} = \sum_{j\in J_{s,f}} \hat{y}_j. \tag{37}$$

By (37), we implicitly assume a timescale separation between solving the problems **MP – UCC** and **PS** [2]. Such assumption is justified to some extend by the following observations. Given the configuration $f \in \mathcal{F}$, problem **MP – UCC** can be solved by standard distributed flow control algorithms [2], 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 overhead involved in configuring paths and setting up connections.

With the two timescale separation in place, we focus on solving the combinatorial problem **PS** in the slow timescale.

C. Approach by Markov Approximation

Following similar procedure in Section III, we apply Markov approximation and at the end turn to solve an approximated version of problem **PS** as follows:

$$\begin{aligned} \mathbf{PS-MA}: \quad & \max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \sum_{s\in S} U_s(z_s) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \\ \text{s.t.} \quad & z_s \leq \sum_{f\in\mathcal{F}} R_{s,f}\, p_f \quad \forall s \in S \\ & \sum_{f\in\mathcal{F}} p_f = 1. \end{aligned} \tag{38}$$

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 a similar analysis as in Section III, the optimal solution of the problem in (38) can be obtained by searching the saddle point of the following function

$$\sum_{s\in S} U_s(z_s) - \sum_{s\in S} \lambda_s z_s + \frac{1}{\beta} \log\left[\sum_{f\in\mathcal{F}} \exp\left(\beta \sum_{s\in S} R_{s,f}\, \lambda_s\right)\right], \tag{39}$$

and at the same time setting

$$p_f(\beta\boldsymbol{\lambda}) = \frac{\exp\left(\beta \sum_{s\in S} R_{s,f}\, \lambda_s\right)}{\sum_{f\in\mathcal{F}} \exp\left(\beta \sum_{s\in S} R_{s,f}\, \lambda_s\right)}, \forall f \in \mathcal{F}. \tag{40}$$

We explore algorithm design based on this observation in the following subsections. The $p_f(\beta\boldsymbol{\lambda})$ in (40) 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 (39).

D. Design and Implementation of Markov Chain

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 $s(f, f')$, so 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 extend, we find that a unique structure of our problem can simplify the implementation.

First, we introduce a new concept. 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., $\mathcal{N}(j) = \{j' : j' \cap j \neq \emptyset\}$. Since there is at most one bottleneck link per path, we have 1) only one link of path j is shared with other paths in $\mathcal{N}(j)$; 2) this particular path 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 $j' \notin \mathcal{N}(j)$, $\mathcal{N}(j') \cap \mathcal{N}(j) = \emptyset$.

Then we have the following observation:

Lemma 2: Under the setting of uncoordinate congestion control, 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', i.e.,

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

Proof: Under the setting of uncoordinate 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 (36) and (37), the equilibrium rate $R_{s',f}$ is invariant and equals to $R_{s',f'}$. ■

Let $H(f, f')$ be the set of such "invariant" users under configurations f and f'. Then to satisfy the detailed balance equation $q_{f,f'} p_f(\beta\lambda) = q_{f',f} p_{f'}(\beta\lambda)$ for f and f' that satisfy **C1** and **C2**, it is sufficient to let

$$\begin{cases} q_{f,f'} = \left[\exp\left(\beta \sum_{s \in S - H(f,f')} R_{s,f} \lambda_s\right)\right]^{-1}, \\ q_{f',f} = \left[\exp\left(\beta \sum_{s \in S - H(f,f')} R_{s,f'} \lambda_s\right)\right]^{-1}. \end{cases} \tag{41}$$

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 (41), the user $s(f, f)$ needs to collect the information $R_{s,f}\lambda_s$ from s in $S - H(f, f')$.

Noticed 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 fetch them from the routers when its own packets pass by. The shared routers can be thought as 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 (41) in a distributed manner.

We briefly describe the distributed implementation as follows.

Stag0: Initially, every user s randomly selects D_s paths from its path set J_s.

Stag1: User s randomly selects one path out of its not-in-use $|J_s| - D_s$ paths, and randomly selects one path out of its D_s in-use path. User s then counts down according to a random number and swaps these two paths when the count-down expires. Denote the current configuration as f and the targeting configuration as f'. The random number is generated following an exponential distribution with parameter $D_s\left(|J_s| - D_s\right)\left[\exp\left(\beta \sum_{s' \in S - H(f,f')} R_{s',f}\lambda_{s'}\right)\right]^{-1}$, where $\sum_{s' \in S - H(f,f')} R_{s',f}\lambda_{s'}$ can be acquired in the following way. For all s' in S and $j \in J_{s',f}$, user s' adds a header containing $R_{s',f}\lambda_{s'}$ to data packets before sending them out along path j. Every router on path j records the information of $R_{s',f}\lambda_{s'}$ for every s' whose traffic passing through them. Assuming the reverse direction traffic (e.g., ACK packets) uses the same paths as 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.

Stag2: During the count-down, each user s also continuously senses whether other users sharing links with them undertake a path swapping. This can be done by the users who swap paths leave a one-bit of information at the routers, and all users whose traffic passing by this router can collect this bit of information. If a user s senses a path swapping, it will reset its counter and jump to **Stag1**.

Stag3: When user s's count-down expires, it will swap the selected two paths, and jump to **Stag1**.

Corresponding pseudocode is shown in Algorithm 1.

Next we establish that the above distributed procedure in fact implements a time-reversible Markov chain with stationary distribution in (40).

Proof: By C1 and C2, we know that all configurations can reach each other within a finite number of transitions, thus the constructed Markov chain is irreducible. Further, it is a finite state ergodic Markov chain with a unique stationary distribution. We now show that the stationary distribution is indeed (40).

Given the current configuration f, the user $s(f, f')$ chooses f' to be its targeting configuration by random selection,, then it counts down according to a random number that follows an exponential distribution with parameter $D_{s(f,f')}\left(|J_{s(f,f')}| - D_{s(f,f')}\right)\left[\exp\left(\beta \sum_{s' \in S - H(f,f')} R_{s',f}\lambda_{s'}\right)\right]^{-1}$.

Now we want to compute the corresponding transition rate $q_{f,f'}$. We first use a counting argument to compute the probability that user $s(f, f')$ selects the particular targeting path that leads to configuration swapping from f to f'. Given a configuration f, the total number of targeting configurations it can swap to is $\prod_{s' \in S} D_{s'}(|J_{s'}| - D_{s'})$. Among these targeting configurations, the number of targeting configurations is $\prod_{s' \in S - \{s(f,f')\}} D_{s'}(|J_{s'}| - D_{s'})$. Thus the desired probability is

$$\begin{aligned} &\mathbf{Pr}(f \to f' \text{ with user } s(f, f')) \\ &= \frac{\prod_{s' \in S - \{s(f,f')\}} D_{s'}(|J_{s'}| - D_{s'})}{\prod_{s' \in S} D_{s'}(|J_{s'}| - D_{s'})} \quad (42) \\ &= \frac{1}{D_{s(f,f')}(|J_{s(f,f')}| - D_{s(f,f')})} \quad (43) \end{aligned}$$

Upon selecting the particular path, the user $s(f, f')$ performs the path switching with rate $D_{s(f,f')}\left(|J_{s(f,f')}| - D_{s(f,f')}\right)\left[\exp\left(\beta \sum_{s' \in S - H(f,f')} R_{s',f}\lambda_{s'}\right)\right]^{-1}$. Thus overall the transition rate from f to f' is given by the

Algorithm 1

The following procedure runs on each individual user independently. We focus on a particular user s.

procedure INITIALIZATION
  $J_{s,f} \leftarrow D_s$ paths random picked from J_s
  index $\leftarrow 0$
  Invoke Procedure Selection(s)
end procedure

procedure SELECTION(s)
  randomly selects one path $j(s)$ from $J_{s,f}$
  randomly selects one targeting path $j'(s)$ from $J_s - J_{s,f}$
  acquires $\sum_{s' \in S - H(f,f')} R_{s',f} \lambda_{s'}$
  generates a timer

$$T_s \sim \exp\left(D_s \left(|J_s| - D_s \right) \left[\exp\left(\beta \sum_{s' \in S - H(f,f')} R_{s',f} \lambda_{s'} \right) \right]^{-1} \right)$$

  and begin counting down
  while the timer T_s does not expire **do**
    if Senses the existence of path swapping activity **then**
      index $\leftarrow 1$
      break
    end if
  end while
  if index $= 1$ **then**
    Terminates current countdown process and invoke Procedure Selection(s)
    index $\leftarrow 0$
  else Invoke Procedure Swap(s, $j(s)$, $j'(s)$)
  end if
end procedure

procedure SWAP(s, $j(s)$, $j'(s)$)
  user s switches from path $j(s)$ to path $j'(s)$
  leaves one bit information at routers along path $j(s)$ and $j'(s)$
end procedure

following equation

$$q_{f,f'} = D_{s(f,f')} \left(|J_{s(f,f')}| - D_{s(f,f')} \right) \left[\exp\left(\beta \sum_{s' \in S - H(f,f')} R_{s',f} \lambda_{s'} \right) \right]^{-1} \times \mathbf{Pr}(f \to f' \text{with user } s(f,f')) \tag{44}$$

$$= \left[\exp\left(\beta \sum_{s' \in S - H(f,f')} R_{s',f} \lambda_{s'} \right) \right]^{-1} \tag{45}$$

With (40), we see that $p_f \cdot q_{f,f'} = p_{f'} \cdot q_{f',f}$, i.e., the detailed balance equations hold. Thus the constructed Markov chain is time-reversible and its stationary distribution is indeed (40) according to Theorem 1.3 and Theorem 1.14 in [20]. ■

E. Solving Problem **PS − MA** *by Running An Primal-dual Algorithm over Markov Chain*

Following a procedure similar to that in Section III-C4, we design a distributed stochastic primal-dual algorithm to pursue the saddle points of the function in (39), on top of the Markov chain implemented in the previous subsection, as follows:

$$z_s(m+1) = \left[z_s(m) + \epsilon(m) \left(U_s'(z_s(m)) - \lambda_s(m) \right) \right]_+, \ \forall s \in S \tag{46}$$

$$\lambda_s(m+1) = \left[\lambda_s(m) - \epsilon(m) \left(\overline{\theta}_s(m) - z_s(m) \right) \right]_+, \ \ \forall s \in S \tag{47}$$

where $\epsilon(m)$ is the step size, $\overline{\theta}_s(m)$ be the average service rate user s actually obtains within the update interval T_m, and T_m is the time interval between the system updating $(\boldsymbol{\lambda}(m-1), \boldsymbol{z}(m-1))$ and $(\boldsymbol{\lambda}(m), \boldsymbol{z}(m))$.

Again, under suitable choices of step sizes and update intervals, we establish the convergence of the stochastic primal-dual algorithm (46)-(47) as follows.

Theorem 3: Assume that $U_s'(0) < \infty, \forall s \in S$, $\max_{s,m} z_s(m) < \infty$ and $\max_{s,m} \lambda_s(m) < \infty$. The stochastic primal-dual algorithm in (46)-(47) converges to the optimal solution of problem **PS − MA** with probability one if the following conditions for step sizes and update intervals hold:

$$\{T_m\} \text{ is non-decreasing with m,} \tag{48}$$

$$\epsilon(m) > 0 \ \forall m, \ \sum_{m=1}^{\infty} \epsilon(m) = \infty, \ \sum_{m=1}^{\infty} \epsilon^2(m) < \infty, \tag{49}$$

$$\sum_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty. \tag{50}$$

Further, the setting $\epsilon(1) = T(1) = 1, \epsilon(m) = \frac{1}{m}, \ T_m = m, \ m \geq 2$ is one specific choice satisfying conditions (48)-(50).

The proof of this theorem 3 is very similar to the proof of theorem 2. We omit details here.

V. CASE 3: CHANNEL ASSIGNMENT IN WIRELESS LANS

A. Settings

Consider a wireless 802.11 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 via 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. 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 i-th AP. Let $\mathcal{F}$ be the set of all feasible f.

Given a configuration f, the wireless stations compete to access the wireless channels according to standard 802.11 protocol. We denote the downlink throughputs 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. The problem of finding the best channel assignment to maximize system-wide utility is as follows:

$$\mathbf{CA}: \quad \max_{f\in\mathcal{F}} \quad \sum_{i=1}^{N} U_i\left(R_i^f\right). \tag{51}$$

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]. Thus, 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.

Let p_f be the percentage of the time that configuration f is activated, i.e., AP i chooses channel f_i. We could reformulate problem **CA** as follows:

$$\mathbf{CA-AVG}: \quad \max_{p\geq 0} \quad \sum_{f\in\mathcal{F}} p_f \sum_{i=1}^{N} U_i\left(R_i^f\right) \tag{52}$$
$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1.$$

We remark that the problem **CA – AVG** is still hard to solve as the number of variables is still combinatorial.

B. Approach by Markov Approximation

We apply Markov approximation and turn **CA – AVG** to the following **CA – MA** optimization problem:

$$\mathbf{CA-MA}: \quad \max_{p\geq 0} \quad \sum_{f\in\mathcal{F}} p_f \sum_{i=1}^{N} U_i\left(R_i^f\right) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f \tag{53}$$
$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1.$$

Its optimal solution is

$$p_f^* = \frac{\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)}{\sum_{f'\in\mathcal{F}} \exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^{f'}\right)\right)}, \forall f \in \mathcal{F} \tag{54}$$

We consider a continuous time-reversible Markov chain that has the stationary distribution given by 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

$$q_{f,f'} = \left[\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)\right]^{-1}. \tag{55}$$

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.

[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.

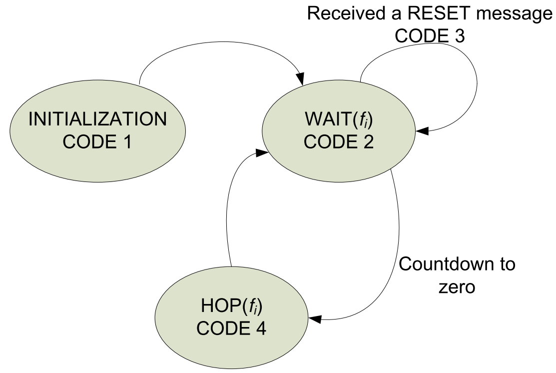


Fig. 2. State machine of a particular link in "Wait-and-Hop" algorithm

C. Implementation

We implement a channel-hopping Markov chain with transition rate in (55) as follows. Initially, the APs randomly pick their channels. Each AP keeps track of its own $U_i(R_i^f)$ based on the measurement of R_i^f under current configuration f, and periodically broadcasts it to all the 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

$$\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)/(M-1) \tag{56}$$

and counts down according to this number. When the count down 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 count down processes and start fresh ones using new measurements under the new configuration f'. We name this implementation "Wait-and-Hop" algorithm for ease of reference.

1) Pseudocode of the "Wait-and-Hop" algorithm: In the "Wait-and-Hop" algorithm, each AP runs a procedure which operates according to the state machine shown in Fig.2. We focus on a particular AP i. The pseudocode under each state is shown in Algorithm 2.

2) Correctness of the "Wait-and-Hop" implementation: We verify that the "Wait-and-Hop" algorithm realizes a continuous-time channel-hopping Markov chain with stationary probability shown in (54).

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.

Denote the probability that the process will enter state f' when leaving state f by $p_{f,f'}$. Let $L = M^N$ be the number of feasible channel assignment states and $\mathcal{N}(f)$ 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, since $|N(f)| = (M-1)N$, we have

$$p_{f,f'} = \frac{1}{|N(f)|} = \frac{1}{(M-1)N}, \forall f' \in \mathcal{N}(f). \tag{57}$$

Algorithm 2 "Wait-and-Hop" algorithm

$C = \{c_1, c_2, \dots c_M\}$
procedure CODE 1
  $f_i \leftarrow$ a channel randomly picked from C
  Transit to State WAIT (f_i)
end procedure

procedure CODE 2
  Measure R_i^f, compute and periodically broadcast $U_i\left(R_i^f\right)$
  Collect $U_j\left(R_j^f\right), j \in \{\{1, \cdots, N\} - \{i\}\}$
  Generate a timer with initial value τ_i that follows exponential distribution with mean equal to $\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)/(M-1)$ and begin counting down
end procedure

procedure CODE 3
  Terminate its current count down 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

In the following, we show the "Wait-and-Hop" implementation realizes a channel-hopping Markov chain with transition rate shown in (55).

- First, all the transition rates of the channel-hopping process are finite;
- Second, $\forall f, f' \in \mathcal{F}$, f and f' communicate with each other. To see this, we transform the state transition diagram of the channel-hopping Markov chain to a directed graph G, in which the vertices represent the channel-assignment states and the arcs represent the transition edges in the state transition diagram, respectively. Now it is sufficient to show that there exists a path on G between f and $f', \forall f, f' \in \mathcal{F}$. Let $m = \mathcal{SP}(f, f')$ denote the length of the shortest path for the process to reach state f' starting from state f. Then we construct a path composed of a sequence of states $f^1, \cdots, f^{m-1}$, such that $\mathcal{SP}(f', f^1) = m-1, \mathcal{SP}(f', f^2) = m-2, \mathcal{SP}(f', f^{m-1}) = 1$. That is, starting from f, the process approaches state f' by changing one element corresponding to state f' upon each transition.
- Third, the transition rate from state f to f' satisfies (55). Given the current state is f, according to the "Wait-and-Hop" algorithm, all N APs count down with rate $(M-1)\left(\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)\right)^{-1}$. Consequently, the rate the process leaves state f is $N(M-1)\left(\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)\right)^{-1}$. With probability $p_{f,f'} = \frac{1}{(M-1)N}$ the process jumps to state f' when leaving state f. Hence, the transition rate from state f to state f' is given by

$$\frac{1}{(M-1)N} \times N(M-1)\left(\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)\right)^{-1} = \left(\exp\left(\beta \sum_{i=1}^{N} U_i\left(R_i^f\right)\right)\right)^{-1}. \quad (58)$$

With (54), we obtain $p_f^* q_{f,f'} = p_{f'}^* q_{f',f}$. That is, the detailed balance equation holds between any two adjacent states. According to [20, Theorem 1.2], the constructed Markoc Chain is time-reversible and its stationary distribution is (54).

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 assignment states.

1) Simulation Setup: 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 simulation run, we gather the statistics of two metrics: i) normalized aggregate throughput; ii) system utility. We define ΔT as the ratio between the achieved normalized aggregate throughput and the optimal normalized aggregate throughput. We further define Utility gap ΔU as the difference between the system utility achieved and the optimal utility.

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

a) 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. In this way, each AP obtains half of the normalized throughput. 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.

b) 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 -0.002.

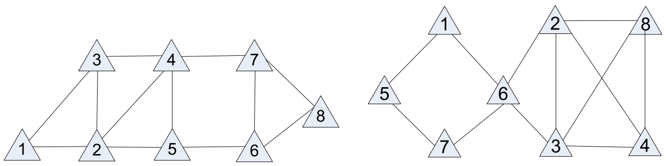
TABLE I
NORMALIZED THROUGHPUT AND UTILITY GAP OF THE "WAIT-AND-HOP" ALGORITHM IN A SIX-AP FULL CLIQUE NETWORK.

Link No.	1	2	3	4	5	6	ΔU
Wait-and-Hop	0.543	0.543	0.543	0.543	0.542	0.541	-0.001

For further details, we show two of these ten networks in Fig.3. Network 1 in Fig.3(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.3(b)

TABLE II
NORMALIZED THROUGHPUT AND UTILITY GAP 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	-0.001	-0.002	-0.001	-0.001	-0.002	-0.003	-0.001	-0.002	-0.003	-0.002	-0.002
ΔT	99.87%	99.85%	99.90%	99.88%	99.84%	99.80%	99.78%	99.85%	99.85%	99.89%	99.85%



(a) Network 1 (b) Network 2

Fig. 3. Two random eight-AP networks

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 Network 1 and Network 2 given in Table III.

TABLE III
NORMALIZED THROUGHPUT OF "WAIT-AND-HOP"IN TWO RANDOM EIGHT-AP NETWORKS IN FIG.3.

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

As shown in Table III, "Wait-and-Hop" can achieve 99.38% and 99.71% of the optimal aggregate throughputs for Network 1 and Network 2, respectively. The utility gaps of "Wait-and-Hop" for Network 1 and Network 2 are -0.002 and -0.001, respectively.

3) Utility loss bound: Eqn. (4) provides a performance loss bound for our Markov approximation. In the worst case, the Log-sum-exp approximation can incur a performance loss of $\frac{1}{\beta}\log n$, where n is the number of feasible configurations. In our simulation setup, we have

$$\frac{1}{\beta}\log n = \begin{cases} \frac{1}{10}\log 3^6 = 0.6592 \text{ for Six} - \text{AP clique network} \\ \frac{1}{10}\log 3^8 = 0.8789 \text{ for Eight} - \text{AP random networks} \end{cases}$$

Simulation results presented in Table I and Table II show that "Wait-and-Hop" can achieve a utility loss of 0.001 and 0.002 for the Six-AP full clique network 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.

VI. CONCLUSIONS

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 value of a combinatorial problem gives rise to a solution that can be realized by time-reversible Markov chains. These Markov chains usually have desirable structure 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 the domain of CSMA networks. This example offers a fresh perspective to a known distributed algorithm. 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: 1) optimal path selection in multipath transmission; 2) frequency channel assignment to WLANs. Based on the promising results out of our preliminary investigation, we believe the Markov approximation techniques will also find application in many network optimization problems in other domains.

APPENDIX

A. Proof of the 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 (12). 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

$$q_{f,f'} = \alpha \left[\exp\left(\beta \sum_{r \in R} x_r(f)\right)\right]^{-1}. \tag{59}$$

Thus Y is an ergodic Markov chain with unique stationary distribution. By (59) and (12), we can check that detailed balance equations hold, by Theorem 1.3 and Theorem 1.14 in [20], we know that this Y is reversible, and its stationary distribution is indeed $p_f^*(\boldsymbol{x})$ in (12).

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 (12). 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 ($1 \leq i \leq |V| - 1$), 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 $v^j_{0,i}(0 \leq j \leq m_i)$. Then according to detailed balanced equations along the PATH(0, i), we have the following:

$$\pi_i = \pi_0 \cdot \prod_{j=0}^{m_i-1} \rho_{v^j_{0,i}, v^{j+1}_{0,i}}, \quad 1 \leq i \leq |V| - 1 \tag{60}$$

$$\pi_0 + \sum_{i=1}^{|V|-1} \pi_i = 1 \tag{61}$$

Then we get the distribution

$$\pi_0 = \frac{1}{1 + \sum\limits_{k=1}^{|V|-1} \prod\limits_{j=0}^{m_k-1} \rho_{v^j_{0,k}, v^{j+1}_{0,k}}} \tag{62}$$

$$\pi_i = \frac{\prod_{j=0}^{m_i-1} \rho_{v^j_{0,i}, v^{j+1}_{0,i}}}{1 + \sum\limits_{k=1}^{|V|-1} \prod\limits_{j=0}^{m_k-1} \rho_{v^j_{0,k}, v^{j+1}_{0,k}}}, \quad 1 \leq i \leq |V| - 1 \tag{63}$$

We now verify the distribution computed based on the spanning tree , i.e., (62)-(63), 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 (63), 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 $j' = v^0_{j',j}$, 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 [20], we have

$$\rho_{j',j} = \prod_{k=0}^{m_{j',j}-1} \rho_{v^k_{j',j}, v^{k+1}_{j',j}} \tag{64}$$

By (63) and (64) we have

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

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 (62)-(63) is indeed the stationary distribution.

Further, the stationary distribution π shown in (62)-(63) can be expressed in the product form in (12) as follows:

$$\pi_0 = \frac{\exp(0)}{\exp(0) + \sum\limits_{k=1}^{|V|-1} \exp\left(\sum\limits_{j=0}^{m_k-1} \log \rho_{v^j_{0,k}, v^{j+1}_{0,k}}\right)} \tag{66}$$

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

So we get the desired conclusion. ■

B. *Proof of the Theorem 2*

Before the further illustration, we need some notation. The vector $\mathbf{z}$ 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}, $m = 0, 1, 2, \ldots$.. $\mathbf{z}(t), \boldsymbol{\lambda}(t)$ remain the same in period m. Let $\mathbf{z}(m), \boldsymbol{\lambda}(m)$ be the value of $\mathbf{z}(t), \boldsymbol{\lambda}(t)$ for all $t \in [t_m, t_{m+1})$. To begin with, we assume $\mathbf{z}(0) = \mathbf{1}$ and $\boldsymbol{\lambda}(0) = \mathbf{0}$ for simplicity. Let $\theta_l(m) = \sum_{l \in f} p_{\underline{f}}(\beta \boldsymbol{\lambda}(m))$ 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

$$L_\beta(\mathbf{z}, \boldsymbol{\lambda}) = \sum_{s \in S} U_s(z_s) - \sum_{l \in L} \lambda_l \sum_{s: l \in s, s \in S} z_s + \frac{1}{\beta} \log\left[\sum_{f \in \mathcal{F}} \exp\left(\beta \sum_{l \in f} \lambda_l\right)\right]. \tag{68}$$

Then the stochastic primal-dual algorithm is given as follows:

$$\begin{cases} z_s(m+1) = \left[z_s(m) + \epsilon(m)\left(U'_s(z_s(m)) - \sum_{l: l \in s} \lambda_l(m)\right)\right]_+ \\ \forall s \in S \quad \text{user rates updating} \\ \lambda_l(m+1) = \left[\lambda_l(m) - \epsilon(m)\left(\bar{\theta}_l(m) - \sum_{s: l \in s} z_s(m)\right)\right]_+ \\ \forall l \in L \quad \text{link prices updating} \end{cases}, \tag{69}$$

Where $\epsilon(m)$ is the step size. 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 one to the optimal solution of $\mathbf{MP} - \mathbf{MA}$ (29). Thus when $\beta \to \infty$, the stochastic primal-dual algorithm (31) (or (69)) converges with probability one to the optimal solution of problem $\mathbf{MP}$ (18).

Now we state the convergence theorem as follows, which is similar to [15, Theorem 7].

Theorem 4: 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:

$$\{T_m\} \text{ is non-decreasing with m} \tag{70}$$

$$\epsilon(m) > 0 \; \forall m, \; \sum\nolimits_{m=1}^{\infty} \epsilon(m) = \infty, \; \sum\nolimits_{m=1}^{\infty} \epsilon^2(m) < \infty \tag{71}$$

$$\sum\nolimits_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty \tag{72}$$

Then by running the stochastic primal-dual algorithm (31), $\boldsymbol{z}$ 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 **MP – MA** (29).

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 (70)-(72). Further, this setting only depends on the index m, and thus can be generally applied to any network.

Thus by theorem 4, we prove that theorem 2 holds.

Now we state the proof of theorem 4, 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 (69) are unbiased, a standard method of stochastic approximation [16]. The difference between our proof and [15] is that, our proof studies the saddle points of Lagrangian function, while [15] 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 $U(m) = (\overline{\boldsymbol{\theta}}(m-1), \boldsymbol{z}(m), \boldsymbol{\lambda}(m), \boldsymbol{y}^0(m))$ 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-filed generated by $U(0), U(1), \ldots, U(m)$, denoted by

$$\mathcal{F}_m = \sigma(U(0), U(1), \ldots, U(m)). \tag{73}$$

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

$$\begin{cases} f_s(m) = U_s'(z_s(m)) - \sum_{l:l\in s} \lambda_l(m), & \forall s \in S \\ g_l(m) = \theta_l(m) - \sum_{s:l\in s} z_s(m), & \forall l \in L \end{cases}, \tag{74}$$

However, in stochastic primal-dual algorithm (31), we only have an estimation of $g_l(m)$ for all $l \in L$, denoted by

$$\overline{g}_l(m) = \overline{\theta}_l(m) - \sum_{s:l\in s} z_s(m), \forall l \in L \tag{75}$$

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

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

$$B_l(m) \triangleq E[\overline{g}_l(m)|\mathcal{F}_m] - g_l(m) = E[\overline{\theta}_l(m)|\mathcal{F}_m] - \theta_l(m) \tag{76}$$

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

$$\eta_l(m) \triangleq \overline{g}_l(m) - E[\overline{g}_l(m)|\mathcal{F}_m] = \overline{\theta}_l(m) - E[\overline{\theta}_l(m)|\mathcal{F}_m] \tag{77}$$

Therefore,

$$\overline{g}_l(m) = g_l(m) + B_l(m) + \eta_l(m), \forall l \in L \tag{78}$$

Remind that $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}})$ is the optimal solution to the problem **MP – MA** (29). 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:

$$V(\boldsymbol{z}, \boldsymbol{\lambda}) \triangleq \| \boldsymbol{z} - \hat{\boldsymbol{z}} \|^2 + \| \boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}} \|^2 \tag{79}$$

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

$$H_\mu \triangleq \{(\boldsymbol{z}, \boldsymbol{\lambda}) : L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}) - L_\beta(\boldsymbol{z}, \hat{\boldsymbol{\lambda}}) \leq \mu\} \tag{80}$$

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

$$L_\beta(\boldsymbol{z}, \hat{\boldsymbol{\lambda}}) \leq L_\beta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\lambda}}) \leq L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}) \tag{81}$$

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

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

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

$$\begin{aligned} z_s(m+1) &= [z_s(m) + \epsilon(m) \cdot f_s(m)]_+, \quad \forall s \in S \\ \lambda_l(m+1) &= [\lambda_l(m) - \epsilon(m) \cdot \overline{g}_l(m)]_+, \quad \forall l \in L \end{aligned}$$

by using the fact that the projection $[\cdot]_+$ is non-expansive [11], we have

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

and with (78), we have

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

Since $U_s'(\cdot)$, $z_s(m)$ and $\lambda_l(m)$ are bounded, by (74) and (75), we know that both $\| \boldsymbol{f}(m) \|^2$ and $\| \overline{\boldsymbol{g}}(m) \|^2$ are bounded, we can write that $\| \boldsymbol{f}(m) \|^2 \leq C_1$ and $\| \overline{\boldsymbol{g}}(m) \|^2 \leq C_2$, where C_1 and C_2 are positive constants. Using this and the above inequalities, we have that

$$\begin{aligned} &V(\boldsymbol{z}(m+1), \boldsymbol{\lambda}(m+1)) \\ &= \| \boldsymbol{z}(m+1) - \hat{\boldsymbol{z}} \|^2 + \| \boldsymbol{\lambda}(m+1) - \hat{\boldsymbol{\lambda}} \|^2 \\ &\leq V(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) + 2\epsilon(m) \cdot [(\boldsymbol{z}(m) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(m) \\ &- (\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}})^T \boldsymbol{g}(m)] - 2\epsilon(m) \cdot [\boldsymbol{\lambda}(m) - \hat{\boldsymbol{\lambda}}]^T [\boldsymbol{B}(m) + \boldsymbol{\eta}(m)] \\ &+ \epsilon^2(m) \cdot (C_1 + C_2) \end{aligned} \tag{82}$$

Assuming that $(\boldsymbol{z}(m), \boldsymbol{\lambda}(m)) \notin H_\mu$ (recall the definition of H_μ in (80)). Then we have

$$L_\beta(\hat{\boldsymbol{z}}, \boldsymbol{\lambda}(m)) - L_\beta(\boldsymbol{z}(m), \hat{\boldsymbol{\lambda}}) \geq \mu \tag{83}$$

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

$$L_\beta(\boldsymbol{z}(m), \lambda(m)) - L_\beta(\hat{\boldsymbol{z}}, \lambda(m)) \geq (\boldsymbol{z}(m) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(m) \quad (84)$$

$$L_\beta(\boldsymbol{z}(m), \hat{\lambda}) - L_\beta(\boldsymbol{z}(m), \lambda(m)) \geq -(\lambda(m) - \hat{\lambda})^T \boldsymbol{g}(m) \quad (85)$$

By the summation of (84) and (85), and combining (83), we have

$$\begin{aligned}&(\boldsymbol{z}(m) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(m) - (\lambda(m) - \hat{\lambda})^T \boldsymbol{g}(m)\\ &\leq L_\beta(\boldsymbol{z}(m), \hat{\lambda}) - L_\beta(\hat{\boldsymbol{z}}, \lambda(m)) \quad (86)\\ &\leq -\mu\end{aligned}$$

Combining with (82) yields that

$$\begin{aligned}&V(\boldsymbol{z}(m+1), \lambda(m+1))\\ &\leq V(\boldsymbol{z}(m), \lambda(m)) - 2\epsilon(m)\mu\\ &+ 2\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(m)]^T [\boldsymbol{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2(m) \cdot (C_1 + C_2) \quad (87)\end{aligned}$$

Further,

$$\begin{aligned}&E[V(\boldsymbol{z}(m+1), \lambda(m+1))|\mathcal{F}_m]\\ &\leq V(\boldsymbol{z}(m), \lambda(m)) - 2\epsilon(m)\mu\\ &+ 2\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(m)]^T [\boldsymbol{B}(m)] + \epsilon^2(m) \cdot (C_1 + C_2) \quad (88)\end{aligned}$$

By lemma 3 and condition (71), $|\Sigma_m\{\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(m)]^T \boldsymbol{B}(m)\}| < \infty$ and $\Sigma_m \epsilon^2(m) \cdot (C_1 + C_2) < \infty$. Then by supermartingale convergence lemma [21], we can conclude that the set H_μ is recurrent for $\{\boldsymbol{z}(m), \lambda(m)\}$.

Step 2: By (82) we have that for $n \geq m+1$,

$$\begin{aligned}&V(\boldsymbol{z}(n), \lambda(n))\\ &\leq V(\boldsymbol{z}(m), \lambda(m)) + 2\sum_{i=m}^{n-1}\{\epsilon(i) \cdot [(\boldsymbol{z}(i) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(i)\\ &- (\lambda(i) - \hat{\lambda})^T \boldsymbol{g}(i)]\} + 2\sum_{i=m}^{n-1}\{\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(i)]^T [\boldsymbol{B}(i) + \boldsymbol{\eta}(i)]\}\\ &+ (C_1 + C_2) \sum_{i=m}^{n-1} \epsilon^2(i) \quad (89)\end{aligned}$$

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

$$\lim_{m\to\infty} (C_1 + C_2) \sum_{i=m}^{\infty} \epsilon^2(i) = 0 \quad (90)$$

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

$$\lim_{m\to\infty} \sum_{i=m}^{\infty} |\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(i)]^T \boldsymbol{\eta}(i)| = 0 \quad (92)$$

Combining (90), (91), and (92), we know that with probability 1, for any $\zeta > 0$, after $(\boldsymbol{z}(m), \lambda(m))$ returns to H_μ for some sufficiently large m (due to recurrence of H_μ),

$$\begin{aligned}&2\sum_{i=m}^{n-1}\{\epsilon(i) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(i)]^T [\boldsymbol{B}(i) + \boldsymbol{\eta}(i)]\}\\ &+ (C_1 + C_2) \sum_{i=m}^{n-1} \epsilon^2(i) \leq \zeta \quad (93)\end{aligned}$$

for any $n \geq m+1$.

Combining (81) and (86), we have that

$$[(\boldsymbol{z}(i) - \hat{\boldsymbol{z}})^T \boldsymbol{f}(i) - (\lambda(i) - \hat{\lambda})^T \boldsymbol{g}(i)] \leq 0 \quad (94)$$

Therefore, applying (93) and (94) to (89), we have

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

Thus $(\boldsymbol{z}(n), \lambda(n))$ can not 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: $\sum_{m=1}^{\infty} |\epsilon(m) \cdot [\hat{\boldsymbol{\lambda}} - \lambda(m)]^T \boldsymbol{B}(m)| < \infty$

The proof is very similar to the one in [15]. It is done by combining two standard methods in Markov chain: bounds on mixing time [18] and uniformization [17]. 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 $\lambda(m)$, 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 l-th 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 (24), $\forall \boldsymbol{y}$, the stationary distribution of state $\boldsymbol{y}$ is

$$\pi_{\boldsymbol{y}}(\lambda(m)) = p_{\boldsymbol{y}}(\beta\lambda) = \frac{\exp\left(\beta \sum_{l\in L} y_l \lambda_l\right)}{\sum_{\boldsymbol{y}'} \exp\left(\beta \sum_{l\in L} y'_l \lambda_l\right)} = \frac{\exp\left(\beta \sum_{l\in L} y_l \lambda_l\right)}{C(\lambda(m))}, \quad (95)$$

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

Since $\lambda(m) \geq \boldsymbol{0}$, $C(\lambda(m)) \leq \sum_{\boldsymbol{y}'} \exp(\beta \boldsymbol{1}^T \lambda(m)) \leq 2^{|L|} \exp(\beta \boldsymbol{1}^T \lambda(m))$.

Thus the minimal probability in the stationary distribution

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

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

$$\begin{aligned}\pi_{\min}(\lambda(m)) &\geq \exp(-|L| \cdot \log 2 - \beta |L| \lambda_{\max}\\ &= \exp(-|L| \cdot (\log 2 + \beta \lambda_{\max})) \quad (96)\end{aligned}$$

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 [17] 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

$$v_m = |L| \cdot \exp(\beta \lambda_{\max}). \quad (97)$$

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 [17], *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 $t_m + t$ $(0 \le t \le T_m)$, 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 $\lambda(m)$. Let $\boldsymbol{y}(t_m + t)$ be the state at time $t_m + t$, then

$$\begin{aligned} & E[\bar{\theta}_l(m)|\mathcal{F}_m]] \\ &= E[\int_0^{T_m} 1 \cdot I(y_l(t_m + t) = 1)dt/T_m] \\ &= \int_0^{T_m} 1 \cdot P(y_l(t_m + t) = 1)dt/T_m \\ &= \int_0^{T_m} E[y_l(t_m + t)|\mathcal{F}_m]dt/T_m \\ &= \int_0^{T_m} \sum_{\boldsymbol{y}'} y' \cdot \omega_m(t, \boldsymbol{y}')dt/T_m \\ &= \sum_{\boldsymbol{y}'} y' \cdot \int_0^{T_m} \omega_m(t, \boldsymbol{y}')dt/T_m \\ &= \sum_{\boldsymbol{y}'} y' \cdot \bar{\omega}_m(\boldsymbol{y}') \end{aligned} \tag{98}$$

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 $\pi_{\boldsymbol{y}^0}(\lambda(m))$ be the probability of $\boldsymbol{y}^0(m)$ in the stationary distribution of *Y(t)*. Let $\boldsymbol{\pi(\lambda(m))} \triangleq \{\pi_{\boldsymbol{y}}(\lambda(m)\}$ be the stationary distribution of *Y(t)*, then by uniformization theorem [17], $\boldsymbol{\pi(\lambda(m))}$ is also the stationary distribution of *Z(n)*.

We use $||\cdot||_{TV}$ to denote the total variation distance between two distributions [18], 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 [18]:

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

Therefore,

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

Further,

$$\begin{aligned} & ||\bar{\boldsymbol{\omega}}_m - \boldsymbol{\pi(\lambda(m))}||_{TV} \quad (99) \\ &= ||\int_0^{T_m} [\boldsymbol{\omega}_m(t) - \boldsymbol{\pi(\lambda(m))}]dt/T_m||_{TV} \\ &\le \int_0^{T_m} ||\boldsymbol{\omega}_m(t) - \boldsymbol{\pi(\lambda(m))}||_{TV} dt/T_m \\ &\le \frac{1}{2}\sqrt{\frac{1}{\pi_{\min}(\lambda(m))}} \frac{1}{v_m(1 - \rho_2)T_m} \quad (100) \end{aligned}$$

Now we bound ρ_2 by Cheeger's inequality [18]

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

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

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

Here Ω is the state space, $\pi_N(\lambda(m)) = \Sigma_{\boldsymbol{y} \in N} \pi_{\boldsymbol{y}}(\lambda(m))$ and $F(N, N^c) = \Sigma_{\boldsymbol{y} \in N, \boldsymbol{y}' \in N^c} \pi_{\boldsymbol{y}}(\lambda(m))P(\boldsymbol{y}, \boldsymbol{y}')$.

Thus

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

then

$$\frac{1}{1 - \rho_2} \le \frac{2}{\phi^2} = 2 \cdot v_m^2 [\pi_{\min}(\lambda(m))]^{-2}. \tag{101}$$

Combing (101), (96) , (97) with (100), it follows that

$$\begin{aligned} & ||\bar{\boldsymbol{\omega}}_m - \boldsymbol{\pi(\lambda(m))}||_{TV} \\ &\le \frac{v_m}{T_m}[\pi_{\min}(\lambda(m))]^{-5/2} \\ &= (|L|/T_m) \cdot \exp[(5/2|L| + 1)\beta\lambda_{\max} + 5/2|L| \log 2] \\ &= (|L| \cdot \tau)/T_m, \end{aligned}$$

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

So by (76) and (98), we have

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

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

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

Therefore

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

where the last step follows from condition (72). ■

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 (73) and (77), 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

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

Thus

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

where the last step follows from condition (71). By Martingale Convergence Theorem [16], $W(n)$ converges with probability 1. ■

References

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

[2] P. Key, L. Massoulie, and P. Towsley, "Path selection and multipath congestion control," in *Proceedings of IEEE INFOCOM*, 2007, pp. 143–151.

[3] S. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-Envelope Computation of Throughput Distributions in CSMA Wireless Networks," *to appear in IEEE Transactions on Mobile Computing. Technical report also available at http://arxiv. org//pdf/0712.1854*.

[4] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proceedings IEEE INFOCOM 2005*, 2005.

[5] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," in *Proceedings of 46th Allerton Conference*, 2008.

[6] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. Poor, "Towards Utility-optimal Random Access Without Message Passing," *Special issue in Wiley Journal of Wireless Communications and Mobile Computing*, Dec, 2009.

[7] S. Rajagopalan and D. Shah, "Distributed algorithm and reversible network," in *Proceedings of CISS*, 2008.

[8] J. Walrand, "Entropy in communication and chemical systems," in *First International Symposium on Applied Sciences on Biomedical and Communication Technologies, (ISABEL'08)*, 2008, pp. 1–5.

[9] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks," in *Proceedings of IEEE INFOCOM Mini-Conference*, 2010.

[10] M. Chiang, *Geometric programming for communication systems*. Now Publishers Inc., 2005.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] P. Laarhoven and E. Aarts, *Simulated annealing: theory and applications*. Springer, 1987.

[13] S. Chau, M. Chen, and S. Liew, "Capacity of Large Scale CSMA Wireless Networks," in *Proceedings of ACM MOBICOM*, 2009.

[14] S. Liu, T. Basar, and R. Srikant, "Controlling the Internet: A survey and some new results," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 3, 2003.

[15] L. Jiang and J. Walrand, "Convergence and Stability of a Distributed CSMA Algorithm for Maximal Network Throughput," Technical Report, UC Berkeley, Mar. 2009. URL: http://www. eecs. berkeley. edu/Pubs/TechRpts/2009/EECS-2009-43. html, Tech. Rep.

[16] H. Kushner and G. Yin, *Stochastic approximation and recursive algorithms and applications*. Springer Verlag, 2003.

[17] M. Kijima, *Markov processes for stochastic modeling*. CRC Press, 1997.

[18] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of Markov chains," *The Annals of Applied Probability*, pp. 36–61, 1991.

[19] C. Cetinkaya and E. Knightly, "Opportunistic traffic scheduling over multiple network paths," in *Proceedings of INFOCOM*, vol. 3, 2004.

[20] F. Kelly, *Reversibility and stochastic networks*. Wiley,Chichester, 1979.

[21] D. Bertsekas and J. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.