A High-order Markov Chain Based Scheduling Algorithm for Low Delay in CSMA Networks

Jaewook Kwak, *Student Member, IEEE,* Chul-Ho Lee, *Member, IEEE,* and Do Young Eun, *Senior Member, IEEE*

***Abstract*—Recently, several CSMA algorithms based on the Glauber dynamics model have been proposed for wireless link scheduling, as viable solutions to achieve the throughput optimality, yet simple to implement. However, their delay performance still remains unsatisfactory, mainly due to the nature of the underlying Markov chains that imposes a fundamental constraint on how the link state can evolve over time. In this paper, we propose a new approach toward better queueing delay performance, based on our observation that the algorithm needs not be Markovian, as long as it can be implemented in a distributed manner. Our approach hinges upon utilizing past state information observed by local link and then constructing a high-order Markov chain for the evolution of the feasible link schedules. We show that our proposed algorithm, named *delayed CSMA*, achieves the throughput optimality, and also provides much better delay performance by effectively 'de-correlating' the link state process (and thus resolves link starvation). Our simulation results demonstrate that the delay under our algorithm can be reduced by a factor of 20 in some cases, compared to the standard Glauber-dynamics-based CSMA algorithm.**

I. Introduction

Medium access control (MAC), which decides link level data transmission, is a central component in wireless packet scheduling. As the MAC plays an important role in achieving efficient channel utilization and providing quality of service for diverse wireless applications, designing an efficient MAC algorithm has been considered to be of significant importance. In a rich and long history of research on this subject, the most commonly believed goal is to achieve the following properties: (i) high throughput, (ii) low delay, and (iii) simple and distributed implementation. These three criteria, however, have different tradeoffs among them, and hence developing an algorithm that has all the properties simultaneously is still a challenging problem.

In wireless networks, the property of high throughput is often determined by the packet arrival rate region under which the algorithm stabilizes the network queues. A classical algorithm, the max-weight scheduling (MWS) [1], is known to achieve the largest rate region under a general independent set constraint model. The MWS, however, is not deemed practical since it requires global information to solve a complicated combinatorial optimization problem in each time instance. Many heuristics such as greedy-maximal scheduling (GMS) or maximal-matching algorithms are considered as alternatives to MWS, but they may achieve only a fraction of the capacity region [2], [3], or are throughput-optimal only on certain types of network topology [4], [5], [6]. There are also several methodologies for achieving the throughput optimality in general network [7], [8], [9], but they have turned out to incur excessive message passing in many cases.

A great advance has recently been made toward this problem in a class of CSMA scheduling, where the throughput optimality can be achieved in a simple and distributed manner, e.g. [10], [11], [12], [13], [14]. The basic idea is based on the so-called Glauber dynamics, which is a Monte Carlo Markov Chain (MCMC) method that often provides an approximate solution to a combinatorial optimization problem. The key enabler in realizing the method for the CSMA scheduling is to achieve a desired probability distribution for the max-weight schedule by locally controlling the CSMA parameters without explicit knowledge of arrival rate or neighboring information. For example, in [10], a distributed algorithm is developed to adaptively choose the CSMA parameter with locally observable information, and the throughput optimality is shown under the time-scale separation assumption (the system converges to its stationary regime quickly enough before adaptation of the CSMA parameters). In another approach [12], the optimality is established without the assumption by taking the parameters as slowly varying function of the queue-size. More general sufficient conditions on the function for the throughput optimality have been studied in [14].

Although the fact that these CSMA algorithms guarantee the throughput optimality is an appealing merit, simulations often demonstrate that they incur large backlogs on the queue, and the resulting delay performance is far from being satisfactory. Thus motivated, in this paper, we mainly focus on improving the (queueing) delay performance without sacrificing the other properties, i.e., high throughput and simple implementation.

A. Related work

In the literature, there have been several attempts to improve the delay performance. For example, in [15], Shah and Shin adopt a coloring operation to the CSMA algorithm so as to achieve a constant-bounded property on the average delay. And, Lotfinezhad and Marbach [16] propose an algorithm called U-CSMA that attempts to resolve link starvation problem by periodically resetting the ongoing schedules to an empty schedule. In these approaches, however, it is assumed that the networks have a polynomial growth structure [15],

This work was supported in part by National Science Foundation under Grant No. CNS-1217341. An earlier version of this paper appeared in the Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Toronto, Canada, 2014.

Jaewook Kwak and Do Young Eun are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC. Chul-Ho Lee is with the Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne, FL. E-mail: {jkwak, clee4, dyeun}@ncsu.edu.

or the topologies are of torus or grid types [16]. It is thus questionable if such an improvement still holds for *any* arbitrary network topology. In [17], Lam et al. show that the delay performance can be improved when multiple channels are available, but such multi-channels may not be present in practice. Huang and Lin [18] propose a virtual multiple channel scheme to maximize the aggregate of utilities of links, while achieving low delay. However, adapting their scheme for stabilizing network queues fed by different feasible, yet unknown, arrival rates may be a non-trivial task, and also their scheme requires an additional signaling among neighbors, which is certainly a new overhead. In [19], Lee et al. propose a suit of generalized versions of Glauber dynamics that all achieve the same stationary distribution and suggest that Metropolis-Hastings algorithm leads to better delay performance, but its improvement is still below the performance gain obtained by our approach, as will be shown later.

B. Our contributions

In this paper, we propose a new approach toward better queueing delay performance, based on our observation that the algorithm needs not be Markovian, as long as it can be implemented in a distributed manner. Our algorithm, termed *delayed CSMA*, updates the next schedule not based on the current status, but on 'several steps back' past state information, thus necessitating high-order Markov chain modeling. This schedule update based on 'delayed' information, somewhat counter-intuitively, provides a significant gain in delay performance by effectively removing the strong correlations that persist in the link state process and thus alleviating link starvation problem.

In particular, we show that our algorithm achieves the throughput optimality, yet provides much better delay performance in the steady state by 'reshaping' the correlation structure to our advantage, while keeping the stationary distribution of the schedules intact. Our extensive simulations show that the delay under our algorithm is smaller than the conventional CSMA algorithm based on the Glauber dynamics by often a factor of 20 over a wide range of scenarios. Our analysis also offers an interesting viewpoint about the role of the mixing time on the delay performance [11], [20] by showing the tradeoff between faster mixing time in the transient phase and smaller correlations in the steady state. In addition, since the main idea behind our algorithm is to decide the next schedule depending on several-steps-back state information, thereby leading to low delay, we expect that our approach can be similarly invoked to improve the delay performance of other scheduling algorithms (modeled by reversible Markov chains) updating the next schedule based on the current status.

It is also important to note that our proposed algorithm is implementable in a completely distributed fashion, without any additional message overhead. Our idea of utilizing past history of channel state can be viewed as simulating multiple channels and use them in a round-robin fashion, based only on *locally* observable information. The idea in [18] takes a similar approach to this in that they also utilize multiple virtually constructed channels, however their algorithm suffers from high communication overhead that can grow with the number of virtual channels used. On the other hand, our approach is much simpler to implement, and yet provides significant improvement. We believe that the zero-overhead feature of our algorithm is better suited to distributed settings in wireless networks, will be a significant appealing merit to the practitioners of network designing.

C. The outline of paper

The rest of this paper is organized as follows. In Section 2, we present our network model and the Glauber dynamics for CSMA algorithms, as well as some definitions on capacity region and the throughput optimality. In Section 3, we first explain our motivation and our own approach, and then show that there indeed exists strong correlations in the link state process (the service process of the queue at the link) under the standard Glauber dynamics based CSMA algorithm. We then construct a class of high-order Markov chains (of order T) and present its distributional and correlation properties in the steady-state. In Section 4, we first show how our algorithm leads to better delay performance, and then prove that our algorithm is also throughput optimal. In Section 5, we discuss the impact of transient dynamics in our algorithm and propose simple modification of our algorithm as a remedy. We also discuss the tradeoff between mixing time and correlation structure under our algorithm. Section 6 presents our extensive simulation results under a various network scenarios, and we conclude in Section 7.

II. Preliminaries

A. Network model

We consider a wireless network with a *conflict graph* $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ is the set of links (transmitter-receiver pair), and $\mathcal{E}$ is the set of edges which represents conflict relationship between links. An edge $(i, j) \in \mathcal{E}$ exists between two links i and j if simultaneous use of the two leads to failure of communications. We define a schedule by $\boldsymbol{\sigma} = (\sigma_v)_{v\in\mathcal{N}} \in \{0, 1\}^{|\mathcal{N}|}$, which represents the set of transmitting links. A link v (or node v in the conflict graph $\mathcal{G}$) is active if it is included in the schedule, i.e., $\sigma_v = 1$, and is inactive if otherwise. Without loss of generality, a link rate is assumed to be a unity capacity, i.e., at most one packet can be transmitted over a link when the link is scheduled. A *feasible* schedule is a set of links that can be active at the same time slot according to the conflict relationship $\mathcal{E}$. Thus, a feasible schedule $\boldsymbol{\sigma}$ should satisfy the independent set constraint i.e., $\sigma_i + \sigma_j \leq 1$ for all $(i, j) \in \mathcal{E}$. We denote by Ω the set of all feasible schedules.

In our model, each link is associated with a queue fed by some exogenous traffic arrivals and serviced when the link is active. We consider that a packet arrives to the queue of link v at each time slot t according to a Bernoulli process $A_v(t)$, i.e., $A_v(t)$, $t = 1, 2, \ldots$ are $i.i.d.$ with $\mathbb{E}\{A_v(t)\} = \eta_v$. Let $\boldsymbol{\eta} = (\eta_v)_{v\in\mathcal{N}}$ be the set of arrival rates to the queues in the network. Let $Q(t) = (Q_v(t))_{v\in\mathcal{N}}$ be the number of packets in the queue at time t. Then the queue dynamics is governed by the following recursion:

$$Q_v(t) = [Q_v(t-1) + A_v(t) - \sigma_v(t)]^+, \quad t \geq 1, \qquad (1)$$

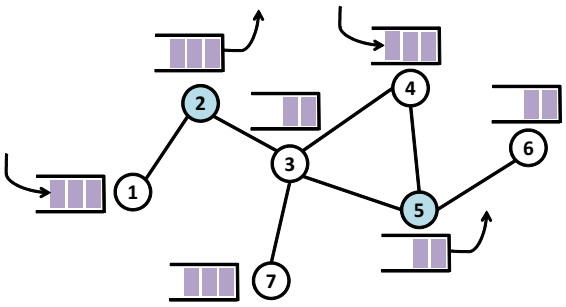


Fig. 1. An instance of schedule where links 2 and 5 are active in a conflict graph with $|\mathcal{N}| = 7$.

where $[x]^+ = \max\{0, x\}$. See Figure 1 for illustration.

B. CSMA scheduling as a Glauber dynamics

The basic idea of throughput-optimal CSMA is to utilize the Glauber dynamics as a link scheduling algorithm. A directly adapted version of the traditional Glauber dynamics to the link scheduling is as follows. First, at every time $t \in \mathbb{N}$, a single link v is chosen uniformly at random from $\mathcal{N}$. We call this link selection procedure as *decision scheduling*. And then, the selected link v updates its state σ_v according to

$$\text{If } \sum_{w \in N_v} \sigma_w(t-1) = 0, \text{ then } \begin{cases} \sigma_v(t) = 1, & \text{w.p. } \frac{\lambda_v}{1+\lambda_v}, \\ \sigma_v(t) = 0, & \text{w.p. } \frac{1}{1+\lambda_v}, \end{cases}$$
$$\text{otherwise, } \sigma_v(t) = 0,$$
$$\text{and for all } w \neq v, \text{ set } \sigma_w(t) = \sigma_w(t-1),$$

where $N_v = \{w : (v, w) \in \mathcal{E}\}$ is a set of neighboring nodes of v, and λ_v is a parameter called *fugacity*. Given $\lambda_v > 0$ for all $v \in \mathcal{N}$, the schedule $\boldsymbol{\sigma}(t)$ forms a Markov chain which is irreducible, aperiodic, and reversible over Ω, and achieves the stationary distribution given by $\pi(\boldsymbol{\sigma}) = \frac{1}{Z} \prod_{i \in \mathcal{N}} \lambda_i^{\sigma_i}$ where $Z = \sum_{\boldsymbol{\sigma} \in \Omega} \prod_{i \in \mathcal{N}} \lambda_i^{\sigma_i}$ is a normalizing constant.

A practical CSMA algorithm uses a modified version of the above procedure. First, the fugacity λ_v can be adjusted to support the link arrival rate. For example, appropriate fugacity parameters can be estimated through experienced arrival and service rate [11], or local queue information [12], [14]. Second, multiple links are allowed to be selected in a single time slot [21], [11] for the decision scheduling. In this procedure, a set of links that do not conflict with each other is selected. This can be achieved in a distributed fashion through a simple randomized procedure. For instance, each link attempts to access the channel with access probability a_v, $v \in \mathcal{N}$, and link v is then selected with probability

$$m_v = a_v \prod_{j \in N_v} (1 - a_j). \tag{2}$$

The set of chosen links $\mathcal{D}(t)$ from this procedure is called a *decision schedule* at time t. More practical implementation tailored to IEEE 802.11 can be found in [21].

It is not difficult to find a continuous time version of the above model. However, the discrete time model has several advantages over the continuous one. For example, by synchronizing the time slots, the well known hidden and exposed terminal problems can be effectively resolved [21]. Also, the continuous time model often requires ideal channel sensing mechanism that may not be feasible in practice [22]. The discrete time model has thus been widely used in the literature, e.g., [11], [14], [22], [21], [18]. We also consider the discrete time model throughout the paper.

C. Capacity region and stability

The capacity region of the network is the set of all arrival rates $\boldsymbol{\eta}$ for which there exists a scheduling algorithm that can support the arrivals. It is known [1] that the region is given by the convex hull of all feasible schedules, i.e.,

$$\mathbb{C} = \left\{ \sum_{\boldsymbol{\sigma} \in \Omega} \theta_{\boldsymbol{\sigma}} \boldsymbol{\sigma} : \sum_{\boldsymbol{\sigma} \in \Omega} \theta_{\boldsymbol{\sigma}} = 1, \theta_{\boldsymbol{\sigma}} \geq 0, \forall \boldsymbol{\sigma} \in \Omega \right\}$$

We first collect several definitions. Let $W_v(t)$ be a weight function associated with a link $v \in \mathcal{N}$ at time slot t. It was shown in [1] that MWS is throughput-optimal with $W_v(t) = Q_v(t)$ (see below for definition in more general settings), provided that it can select a maximum-weight schedule $\boldsymbol{\sigma}^*(t)$ in every time slot where

$$\boldsymbol{\sigma}^*(t) = \arg\max_{\boldsymbol{\sigma} \in \Omega} \sum_{v \in \boldsymbol{\sigma}} W_v(t).$$

This has been generalized in [23] as follows. For all $v \in \mathcal{N}$, set link weights as $W_v(t) = h(Q_v(t))$ for some monotone increasing functions $h : [0, \infty) \to [0, \infty)$. (See [23] for precise definitions for the weight functions $h(\cdot)$.) In this case, a scheduling algorithm is said to be *throughput-optimal* if for all arrival rates inside the capacity region, network queues are stable in the sense that

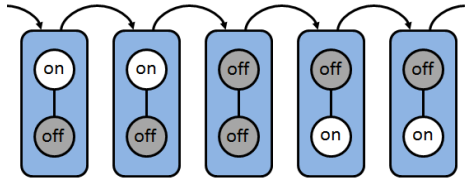
$$\limsup_{K \to \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\left[\left(\sum_{v \in \mathcal{N}} h^2(Q_v(t)) \right)^{1/2} \right] < \infty. \tag{3}$$

When the whole system including all the queue-lengths is a Markov chain, the condition in (3) implies that the chain is positive recurrent [24], [14], [13]. It is however, worth mentioning that the condition in (3) itself is established under a fairly general case in that the system of $Q_v(t)$ doesn't need to be a Markov chain.

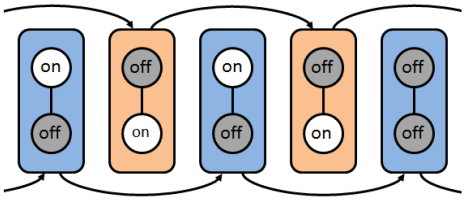
III. CSMA Scheduling Based on High-order Markov Chain

A. Motivation and our approach

According to the standard queueing theory, the queueing delay is governed by not only the long-term average arrival and service rates, but also their higher-order statistics such as their correlations or dependency over time [25], [26]. Indeed, there are a number of works in the literature suggesting that positive correlations have an adverse impact on the queueing delay [27], [28], [29], [30]. With this in mind, in this paper, we aim at developing a new, distributed algorithm similar to the current CSMA-based ones [12], [22], [15], [21], [13], [19], but offering far superior delay performance by effectively reducing such correlations in any general network topology, while keeping the throughput optimality intact.



(a) Conventional CSMA algorithm



(b) Proposed approach: Delayed CSMA

Fig. 2. Comparison between the conventional CSMA and our proposed approach. A box indicates a schedule, and arrows indicate state transitions.

Our motivation comes from the fact that the service process $\sigma_v(t)$ at link v under the standard CSMA policy is often heavily correlated over time. (More detailed discussion is in Section III-B.) This is because once CSMA finds a schedule, it tends to stay in its similar set of schedules for a long time [16]. To illustrate, consider for example two links that interfere with each other, so that only one link can be active at a time. At a particular moment, if a schedule of the two links is 'active-inactive', the inactive link first has to wait until the active link releases the channel occupation. In this case, transition to next possible state is limited to 'active-inactive' or 'inactive-inactive' state, and direct transition to 'inactive-active' state is impossible. (See Figure 2(a).) This phenomenon hinders frequent switch between schedules, leading to starvation for the corresponding inactive link.

The method we propose here effectively resolves this problem. The main idea is as follows. Suppose we have two schedulers that respectively generate schedules independently, while preserving the feasibility constraint for each time slot. If we choose to use one scheduler at every odd time index, and the other one at every even time index (see Figure 2(b)), it is now possible to make a transition from 'active-inactive' directly to 'inactive-active' state, which would be impossible under the conventional CSMA. This alternate use of different schedulers produces more drastic change of states in consecutive time slots, thereby alleviating link starvation while maintaining the same long-term frequency of being active.

Given the potential benefits, we generalize this idea to the use of multiple schedulers in a round-robin manner. Throughout the paper, we will use a notation T to indicate the number of such schedulers. In practice, this can be easily implemented in a distributed setting by having all links together update their schedules based on T-step-back state. For this purpose, each link only needs to remember its last T channel states. This way, the whole system behaves *as if* there are T separate schedulers (or chains) taking turns to generate next schedules. Building upon this idea and applying to the CSMA scheduling, we next present our proposed algorithm, named *delayed CSMA*. Note that if $T = 1$, our algorithm reduces to the conventional CSMA-based scheduling algorithm.

Algorithm 1 Delayed CSMA

Initialize: for all links $i \in \mathcal{N}$, $\sigma_i(t) = 0$, $t = 0, 1, \ldots, T-1$.
At each time $t \geq T$: links find a decision schedule, $\mathcal{D}(t)$ through a randomized procedure, and
for all links $i \in \mathcal{D}(t)$ **do**
  if $\sum_{j \in N_i} \sigma_j(t-T) = 0$ **then**
    $\sigma_i(t) = 1$ with probability $\frac{\lambda_i}{1+\lambda_i}$
    $\sigma_i(t) = 0$ with probability $\frac{1}{1+\lambda_i}$
  else
    $\sigma_i(t) = 0$
  end if
end for
for all links $j \notin \mathcal{D}(t)$ **do**
  $\sigma_i(t) = \sigma_i(t-T)$
end for

B. Understanding correlation structure of the standard CSMA

Before proceed to explain the details about our algorithm, we first study how much correlations are present in the service process $\sigma_v(t)$ for the queue at each link, induced by the standard Glauber dynamics.

To set the stage, consider a homogeneous Markov chain $\boldsymbol{\sigma}(t) \in \Omega$ denoting a feasible configuration by the Glauber dynamics at time slot t, assuming fugacity parameter λ_v is set to be a constant. Since we are interested in the long-run behavior of queueing performance, without loss of generality, we can assume that the Markov chain $\boldsymbol{\sigma}(t)$ is in its stationary regime, i.e., $\mathbb{P}\{\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}\} = \pi(\boldsymbol{\sigma})$, for all $t \geq 0$. We write $\pi(B_v) \triangleq \sum_{\boldsymbol{\sigma} \in B_v} \pi(\boldsymbol{\sigma}) = \mathbb{P}_\pi\{\sigma_v(t) = 1\}$ for the long-term proportion of service availability at link v, where $B_v \triangleq \{\boldsymbol{\sigma} \in \Omega : \sigma_v = 1\} \subseteq \Omega$ is a set of all feasible schedules for which v is active. For the rest of the paper, we mostly focus on the service process and queueing dynamics at a given link $v \in \mathcal{N}$, so we will drop the subscript v and write $\sigma(t)$ for $\sigma_v(t)$ (similarly η for η_v, $A(t)$ for $A_v(t)$, and $Q(t)$ for $Q_v(t)$) unless otherwise necessary.

We first state the following lemma which is useful in understanding the correlation structure of the service process $\sigma(t)$ under Glauber dynamics. An analogous statement for continuous time version was given by Lemma 3.6 of [31], and we reproduce it here for a discrete-time case.

Lemma 1. *[32], [31] Let X_k be an irreducible, aperiodic, and reversible Markov chain with its transition probability matrix P and stationary distribution π. If X_0 is drawn from π, then for $B \subset \Omega$, there exist $\alpha_j \geq 0$, $j = 1, \ldots, |\Omega|$ such that*

$$\mathbb{P}\{X_k \in B | X_0 \in B\} = \sum_{j=1}^{|\Omega|} \alpha_j \rho_j^k, \tag{4}$$

where $\sum_j \alpha_j = 1$, $\alpha_1 = \pi(B)$, and $1 = \rho_1 > \rho_2 \geq \cdots \geq \rho_{|\Omega|} > -1$ are eigenvalues of P.

Let $\psi(t,k) = \mathrm{Cov}\{\sigma(t), \sigma(t+k)\}/\mathrm{Var}\{\sigma(t)\}$ be the Pearson's correlation coefficient of lag k at time t for the service process. Under the stationarity assumption, one can characterize the correlation based only on the lag parameter k,

and hence we will use $\psi(k)$ instead to indicate the correlation of lag k at some time t in the steady state, which is equivalent to saying $\psi(k) = \lim_{t\to\infty}\psi(t,k)$. In the following, we will simply call correlation at lag k to indicate $\psi(k)$. We have the following proposition that characterizes the correlation of the service process.

Proposition 1. *Let $q_v = \sum_{\boldsymbol{\sigma}:\sigma_j=0,\forall j\in N_v}\pi(\boldsymbol{\sigma})$ be the probability that none of neighboring links of v is active. Then,*

$$\psi(1) = 1 - \frac{m_v}{1+(1-q_v)\lambda_v}, \tag{5}$$

$$\psi(2k) \geq \left(1 - \frac{m_v(2-m_v)}{1+(1-q_v)\lambda_v}\right)^k, \quad k = 1,2,\ldots. \tag{6}$$

where m_v is the probability that link v is selected in a decision schedule as in (2).

Proof: Let $E_t = \{\boldsymbol{\sigma}\in B_v\} = \{\sigma_v(t)=1\}$ be the event that link v is scheduled (active) at time t. Clearly, $\mathbb{P}\{E_t\} = \mathbb{E}\{\sigma_v(t)\} = \pi(B_v)$. First, observe that

$$\begin{aligned}\mathrm{Cov}\{\sigma(t),\sigma(t+k)\} &= \mathbb{P}\{E_t, E_{t+k}\} - \pi(B_v)^2\\ &= \pi(B_v)\left(\mathbb{P}\{E_{t+k}|E_t\} - \pi(B_v)\right),\end{aligned}$$

and $\mathrm{Var}\{\sigma(t)\} = \pi(B_v)(1-\pi(B_v))$. Under E_t, link v is active, thus all its neighbors must be inactive. In this situation, there are two possible events for link v to stay active in the next time slot: (a) link v is selected for update, but keep active state, (b) link v is not selected. The probability of event (a) is $\frac{m_v\lambda_v}{1+\lambda_v}$, and that of event (b) is $1-m_v$. Thus,

$$\mathbb{P}\{E_{t+1}|E_t\} = \frac{m_v\lambda_v}{1+\lambda_v} + (1-m_v) = 1 - \frac{m_v}{1+\lambda_v}.$$

In [11], it was shown that $\pi(B_v) = \frac{\lambda_v}{1+\lambda_v}q_v$. (See Appendix B of [11].) Rearranging the terms, (5) follows.

Now consider the probability that if a link v is active, it is also active after two time slots, i.e., $\mathbb{P}\{E_{t+2}|E_t\}$. Let $E_t^c = \{\boldsymbol{\sigma}\notin B_v\}$ denotes an event that link v is inactive at time t. Then,

$$\mathbb{P}\{E_{t+2}|E_t\} = \mathbb{P}\{E_{t+2},E_{t+1}|E_t\} + \mathbb{P}\{E_{t+2},E_{t+1}^c|E_t\}.$$

A simple calculation reveals that $\mathbb{P}\{E_{t+2},E_{t+1}|E_t\} = (1-\frac{m_v}{1+\lambda_v})^2$, which is the probability that the link remains active for the next two consecutive time slots given that it is active now. Similarly, $\mathbb{P}\{E_{t+2},E_{t+1}^c|E_t\}$ is the probability that the link v, from active state, changes to inactive and then active again. The first transition occurs with probability $\frac{m_v}{1+\lambda_v}$, and at this time note that none of the neighbors of v is in the decision schedule. Thus, the second transition will occur with probability $\frac{m_v\lambda_v}{1+\lambda_v}$. Thus, $\mathbb{P}\{E_{t+2}|E_t\} = 1 - \frac{m_v(2-m_v)}{1+\lambda_v}$, and we can write

$$1 - \frac{m_v(2-m_v)}{1+\lambda_v} - \pi(B_v) = \mathbb{P}\{E_{t+2}|E_t\} - \mathbb{P}\{E_t\} = \sum_{j=2}^{|\Omega|}\alpha_j\rho_j^2,$$

where the second equality is from Lemma 4 and by choosing $B = B_v$ and P as the transition probability matrix of the standard Glauber dynamics (Algorithm 1 with $T=1$). Note that

$$\mathbb{P}\{E_{t+2k}|E_t\} - \mathbb{P}\{E_t\} = \sum_{j=2}^{|\Omega|}\alpha_j\rho_j^{2k} = (1-\alpha_1)\frac{1}{1-\alpha_1}\sum_{j=2}^{|\Omega|}\alpha_j\rho_j^{2k}.$$

Define a random variable $Y\geq 0$ which takes value ρ_j^2 with probability $\frac{\alpha_j}{1-\alpha_1}$, $j=2,3,\ldots,|\Omega|$. Then the above can be written as $(1-\alpha_1)\mathbb{E}\{Y^k\}$. From Jensen's inequality, we have

$$(1-\alpha_1)\mathbb{E}\{Y^k\} \geq (1-\alpha_1)\mathbb{E}\{Y\}^k,$$

where the RHS is equal to $(1-\alpha_1)\left(1-\frac{m_v(2-m_v)}{(1-\alpha_1)(1+\lambda_v)}\right)^k$. This proves (6) by noting that $\alpha_1 = \pi(B_v) = \frac{\lambda_v}{1+\lambda_v}q_v$. ■

Note that the obtained lower bounds are all positive since $0\leq m_v\leq 1$. The lower bound is, in general, valid only for even lags, however, it has been shown in [33] that for any finite state reversible Markov chain, the sum of adjacent pairs of correlations, $\psi(2n)+\psi(2n+1)$, is positive, decreasing, and convex in $n\geq 1$. Clearly, the standard Glauber dynamics is a reversible Markov chain on Ω. This implies that, in conjunction with $\psi(1)$ being positive, the negative correlations in $\sigma(t)$, even if they exist, will be of much less magnitude and have lesser impact than positive ones. In addition, it has been shown that the correlations of a general Gibbs sampler with unitary random scan, i.e., updating a single node at a time, are all positive and decrease monotonically [34]. Evidently, the Glauber dynamics is a special instance of Gibbs sampler, so we expect that those results about positive correlation structure can be naturally extended to the case with multiple update scheme as well. For instance, Figure 3(a) shows measured correlations $\psi(k)$ for link $v=3$ in the network topology given in Figure 1, based on the standard Glauber dynamics with access probability $a_i=0.25$ and fugacity $\lambda_i=1$ for all $i\in\mathcal{N}$, along with the predicted lower bounds in (5) and (6). We observe that the degree of correlations in the service process is significant and the lower bound in Proposition 1 is in fact quite tight over a wide range of lags.

C. High-order Markov chain model

The key feature of our algorithm over the conventional one is that a new schedule is generated not from the current schedule but from T-step-back past schedule, and in doing so, each link just needs to make a decision based on its own T-step-back channel state. Since every link operates this way, the independent set constraint is satisfied all the time. From an analytical point of view, however, this means that the evolution of schedule $\boldsymbol{\sigma}(t)$ is no longer a Markov chain on Ω, rendering all the results associated with Markov chains inapplicable.

Notwithstanding being non-Markov, our algorithm can still be modeled by a high-order Markov chain of order T on the same state space Ω, as follows:

$$\begin{aligned}&\mathbb{P}\{X_t = x \mid X_{t-1}=x_{t-1},\ldots,X_0=x_0\}\\ &= \mathbb{P}\{X_t = x \mid X_{t-1}=x_{t-1},\ldots,X_{t-T}=x_{t-T}\},\end{aligned}$$

implying the current state depends upon T past history.[1] Our algorithm can then be written as

$$\mathbb{P}\{X_t = y \mid X_{t-T} = x\} = P(x,y), \tag{7}$$

where $P(x,y)$, $x,y \in \Omega$ is the transition probability of the conventional Markov chain from state x to y.

We here collect several notations and simple facts that hold for finite state Markov chains [32], [35], [36] and will prove useful throughout the analysis. Consider a finite-state, ergodic Markov chain Y_n with its transition matrix P. For functions $f, g : \Omega \to \mathbb{R}$, we define

$$\langle f, \mathbf{P}^{(k)} g\rangle_\mu \triangleq \sum_{x,y\in\Omega} f(x)g(y)\mu(x)P^{(k)}(x,y) = \mathbb{E}_\mu\{f(Y_0)g(Y_k)\},$$

where μ is a probability distribution of Y_0 on Ω, and $P^{(k)}(x,y)$ is k-step transition probability of the chain Y_n. For simplicity, we write $\langle f\rangle_\mu \triangleq \langle f, 1\rangle_\mu = \mathbb{E}_\mu\{f(Y_0)\}$. Also define

$$(\mathbf{P}^{(k)} f)(x) \triangleq \sum_{y\in\Omega} P^{(k)}(x,y)f(x) = \mathbb{E}\{f(Y_k)|Y_0 = x\}. \tag{8}$$

For a (high-order) Markov chain X_t with order T as given in (7), if we define $Y_n^m = X_{nT+m}$ for $0 \le m \le T-1$ and $n = 0, 1, 2, \ldots$, then $\{Y_n^m\}_{n\ge 0}$ for each m is a conventional Markov chain with initial state $Y_0^m = X_m$. Since the chain P is ergodic, we have, for any initial state x,

$$\lim_{k\to\infty} (\mathbf{P}^{(k)} f)(x) = \lim_{k\to\infty} \mathbb{E}\{f(Y_k^m)|Y_0^m = x\} = \langle f\rangle_\pi, \tag{9}$$

where π is the stationary distribution of the chain P. Since this holds for any given $m = 0, 1, \ldots, T-1$, it follows that

$$\lim_{t\to\infty} \mathbb{E}\{f(X_t)|X_0 = x_1, \ldots, X_{T-1} = x_{T-1}\} = \langle f\rangle_\pi,$$

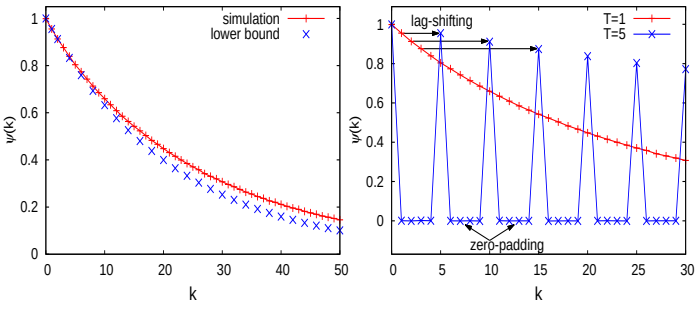
i.e., the marginal distribution of X_t in the steady-state remains the same and does not change with T. Thus, our delayed CSMA algorithm achieves the same stationary distribution as the standard CSMA algorithm. However, as we describe in the following propositions, different T leads to strikingly different behavior in the second order statistics.

Proposition 2. *(Asymptotic zero-correlation padding) Let X_t be a high-order Markov chain of order T where the transition kernel is given by (7). For any initial distribution, if $k \neq lT$, where $l \in \mathbb{N}$, then $\lim_{t\to\infty} \mathbb{E}\{f(X_t)g(X_{t+k})\} = \langle f\rangle_\pi \langle g\rangle_\pi$, assuming that the expectations exist.*

Proof: Writing $t = nT + m$, and $t + k = n'T + m'$ for $m, m' \in \{0, \ldots, T-1\}$ and $n, n' = 0, 1, \ldots$, one can verify that $m \neq m'$ if $k \neq jT$. Define $Y_n^m = X_{nT+m}$, and $Y_{n'}^{m'} = X_{n'T+m'}$. For given m, m', Y_n^m and $Y_{n'}^{m'}$ are both conventional Markov chains with transition kernel P. Then by conditioning, we have

$$\begin{aligned}\mathbb{E}\{f(X_t)g(X_{t+k})\} &= \mathbb{E}\{f(Y_n^m)g(Y_{n'}^{m'})\}\\ &= \mathbb{E}\{\mathbb{E}\{f(Y_n^m)g(Y_{n'}^{m'})|Y_0^m, Y_0^{m'}\}\},\end{aligned}$$

[1] Alternatively, one can also augment the state space into a product space $\Omega\times\cdots\times\Omega$ (T times) on which $\{X_{t-T+1}, \ldots, X_{t-1}, X_t\}$ becomes a Markov chain, but this would lead to largely intractable descriptions.



(a) (b)

Fig. 3. (a): Correlations of $\sigma_3(t)$ for the network in Figure 1 from simulations and the lower bounds. (b): Two properties of the algorithm: correlation-lag shifting and zero-correlation padding.

and observe that

$$\begin{aligned}&\mathbb{E}\{f(Y_n^m)g(Y_{n'}^{m'})|Y_0^m = i, Y_0^{m'} = j\}\\ &= \mathbb{E}\{f(Y_n^m)|Y_0^m = i, Y_0^{m'} = j\}\mathbb{E}\{g(Y_{n'}^{m'})|Y_0^m = i, Y_0^{m'} = j\}\\ &= \mathbb{E}\{f(Y_n^m)|Y_0^m = i\}\mathbb{E}\{g(Y_{n'}^{m'})|Y_0^{m'} = j\}\\ &= (\mathbf{P}^{(n)} f)(i)(\mathbf{P}^{(n')} g)(j),\end{aligned}$$

where the first and second equalities follow from the conditional independence of Y_n^m and $Y_{n'}^{m'}$ when initial values are fixed. Since $n, n' \to \infty$ as $t \to \infty$, by taking limits and from (9), we are done. ∎

Proposition 3. *(Asymptotic correlation-lag shifting) Let X_t be a Markov chain of order T with its transition kernel given by (7). For any initial distribution and for any given $k \in \mathbb{N}$, we have $\lim_{t\to\infty} \mathbb{E}\{f(X_t)g(X_{t+kT})\} = \langle f, \mathbf{P}^{(k)} g\rangle_\pi$, assuming that the expectations exist.*

Proof: As before, write $t = nT + m$ for $m \in \{0, 1, \ldots, T-1\}$ and $n = 0, 1, \ldots$. Then $Y_n^m = X_{nT+m}$ for each m is a Markov chain. Observe that

$$\mathbb{E}\{f(X_t)g(X_{t+kT})\} = \mathbb{E}\{f(Y_n^m)g(Y_{n+k}^m)\} = \langle f, \mathbf{P}^{(k)} g\rangle_{\mu_n^m},$$

where μ_n^m is the distribution of Y_n^m. Since $n \to \infty$ as $t \to \infty$, taking limit gives

$$\lim_{n\to\infty} \langle f, \mathbf{P}^{(k)} g\rangle_{\mu_n^m} = \langle f, \mathbf{P}^{(k)} g\rangle_\pi,$$

since $\mu_n^m \Rightarrow \pi$ as $n \to \infty$ and the state space is finite. This completes the proof. ∎

We consider the functions, $f(\boldsymbol{\sigma}(t)) = g(\boldsymbol{\sigma}(t)) = \mathbf{1}_{\{\boldsymbol{\sigma}(t)\in B_v\}} = \sigma_v(t)$ to indicate the service process of a particular link v. In which case, the expectations in the above propositions always exist, as the state space is finite. Since $\boldsymbol{\sigma}(t)$ under our algorithm is a Markov chain of order T (i.e., modeled as X_t here), Propositions 2 and 3 tell us that the correlations $\psi(k)$ under our delayed CSMA algorithm with order T are first shifted to T times larger lags, and then all padded with zero in-between. To see this effect numerically, we have run the simulations with the same step as in Figure 3(a), but now with $T = 5$. As seen in Figure 3(b), the correlations $\psi(1)$, $\psi(2)$, and $\psi(3)$ for $T = 1$ case respectively gets shifted

to $\psi(5)$, $\psi(10)$, and $\psi(15)$ for $T = 5$, and $\psi(k) = 0$ for all $k \not\equiv 0 \pmod 5$. For the rest of the paper, we call these phenomena *zero-padding* and *lag-shifting*, respectively.

IV. Delay and Throughput Analysis

A. Delay performance

In this section, we investigate the efficiency of our delayed-CSMA algorithm of order T in terms of its delay performance. As discussed earlier, our algorithm provides different second-order behavior while preserving the same long-term average statistics. Our analysis hinges upon the common belief that less variability in the service process generally leads to smaller queue-length and delay when the average statistics remains the same. To this end, we assume for now that the fugacity for each link is a fixed constant. This assumption will be relaxed later in Section IV-C, in which we show our algorithm under any finite T is also throughput optimal when the fugacity can be time-varying, set to be some function of the queue-length at each link. At this point, we focus on the long-term behavior of the delay, as any impact of transient phase will eventually fade away as time goes on, and in the standard queueing literature, the behavior of queue-length or delay is discussed mostly in the steady-state. We will briefly touch upon the impact of transient phase under our algorithm later in Section IV-B.

First, Little's law asserts that the average delay is determined by the average queue length given that arrival rate is kept the same. For this reason, we are here interested in the stationary behavior of the queue-length driven by the recursion in (1). Let $I(t) = A(t) - \sigma(t)$ be the *net input* to the queue at time t, with $\mathbb{E}\{I(t)\} = \eta - \pi(B_v) = -\xi < 0$ for the stability of each queue [37]. Let $\mathbf{A}_t = \sum_{k=1}^{t} A(k)$ and $\mathbf{S}_t = \sum_{k=1}^{t} \sigma(k)$ be the cumulative amounts of arrival and service over t slots, respectively. For a constant C such that $C > \xi = \pi(B_v) - \eta > 0$, we define $Z(t) \triangleq A(t) - \sigma(t) + C$, and $\mathbf{Z}_t \triangleq \sum_{k=1}^{t} Z(k)$. Then, the recursion (1) can be rewritten as $Q(t) = [Q(t-1) + Z(t) - C]^+$, i.e., the queue-length evolves as if the arrival process is $Z(t)$ with $\mathbb{E}\{Z(t)\} = C - \xi$ and the service rate is constant C. Thus, the queue-length Q in the steady-state admits the following.

$$\mathbb{P}\{Q > x\} = \mathbb{P}\left\{\sup_{t\geq 0}\left[\mathbf{Z}_t - Ct\right] > x\right\}.$$

Our next step is to note that, by similar conditioning on the initial values, we can generalize Propositions 2 and 3 into the case of multiple random variables, say, $f(X_{t+t_1}), g(X_{t+t_2}), h(X_{t+t_3}), \ldots$ whose time indices are all distinct in modulo T. Since the choice of functions is arbitrary, this implies that T distinct sub-processes defined by $\sigma^{(i)}(n) = \{\sigma(nT+i)\}_{n\geq 0}$, $i = 1, \ldots, T$ are all independent in the steady state, and the entire correlation structure of the original process $\sigma(n)$ carries over to each of the sub-processes $\sigma^{(i)}(n)$. In particular, let $Z^{(i)}(n) = Z(nT+i)$, $n \geq 0$, for each $i = 1, 2, \ldots, T$. From the zero-padding and lag-shifting properties, and since $A(t)$ is $i.i.d.$ over time t and also independent of $\sigma(t)$, it follows that $\{Z^{(i)}(n)\}_{n\geq 0}$, $i = 1, 2, \ldots, T$ are $i.i.d.$ processes, each of which has $\mathbb{E}\{Z^{(i)}(n)\} = C - \xi$ and $\mathrm{Cov}\{Z^{(i)}(n), Z^{(i)}(n+k)\} = \mathrm{Cov}\{Z(n), Z(n+k)\}$ in the steady state. If we consider a cumulative net-input process up to $t = nT$ for some $n \in \mathbb{N}$, then we can decompose $\mathbf{Z}_t$ into $\mathbf{Z}_t = \sum_{i=1}^{T} \mathbf{Z}_\mathbf{t}^{(\mathbf{i})}$ where

$$\begin{aligned}\mathbf{Z}_\mathbf{t}^{(\mathbf{i})} \triangleq \sum_{k=0}^{n-1} Z^{(i)}(k) &= \mathbf{A}_\mathbf{t}^{(\mathbf{i})} - \mathbf{S}_\mathbf{t}^{(\mathbf{i})} + nC \\ &= \sum_{k=0}^{n-1} \left[A(kT+i) - \sigma(kT+i) + C\right].\end{aligned}$$

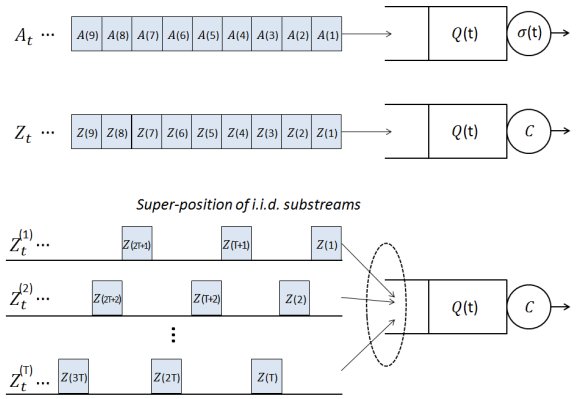


Fig. 4. Our algorithm has an effect of dividing the original net-input process of a single stream (top) into a sum of $i.i.d.$ substreams (bottom) in the steady state.

Thus, the cumulative 'modified net-input' process $\mathbf{Z}_t$ to the queue with constant service rate of C is nothing but a superposition of T $i.i.d.$ replicas of the original ($\mathbf{Z}_t = \sum_{i=1}^{T} \mathbf{Z}_\mathbf{t}^{(\mathbf{i})}$), but with its timescale stretched by T-fold. Our algorithm with parameter T thus effectively de-correlates the original service process (or the modified net-input process), regardless of how much correlations persist in the original case caused by any arbitrary topological constraint under the Glauber dynamics. See Figure 4 for illustration. Also,

$$\mathbb{P}\left\{\sup_{t\geq 0}\left[\mathbf{Z}_t - Ct\right] > x\right\} = \mathbb{P}\left\{\sup_{t\geq 0}\sum_{i=1}^{T}\left[\mathbf{Z}_\mathbf{t}^{(\mathbf{i})} - (C/T)t\right] > x\right\}.$$

In other words, the queue at link v behaves as if we aggregate T $i.i.d.$ input and also aggregate T different service capacities of C/T each. Thus, we expect that the usual benefits of statistical multiplexing gain and the economies of scales [30], [38], [39], or the principle of "Sharing resources is always better than partitioning." [40] apply here. To quantitatively capture such a gain out of de-correlating the original process under our algorithm, we can also employ the usual Gaussian approximation for $\mathbf{Z}_t$, now a sum of T $i.i.d.$ processes, by appealing to the Central Limit Theorem. Note that $\mathbb{E}\{\mathbf{Z}_t\} = (C - \xi)t$ and we define by $v(t, T)$ the variance of $\mathbf{Z}_t$ under our algorithm with order parameter T. Then, it is known that the queue-length distribution with Gaussian input can be well approximated by [41], [42], [25]

$$\mathbb{P}\{Q > x\} \approx \exp\left(-\inf_{t>0}\frac{(\xi t + x)^2}{2v(t,T)}\right)$$

for a wide range of $x > 0$. We then have the following.

Proposition 4. *Suppose the correlations of $\sigma(t)$ under the original CSMA Glauber dynamics are all non-negative, i.e.,*

$\psi(k) \geq 0$ for all $k \in \mathbb{N}$. Then, for any given $t > 0$, $v(t,T)$ is decreasing (non-increasing) in T.

Proof: Let $\boldsymbol{\sigma}(t,T)$ be the configuration state generated from our algorithm, a Markov chain of order T, and set $\sigma(t,T) = \mathbf{1}_{\{\boldsymbol{\sigma}(t,T)\in B_v\}}$ to be the service process of link v under our algorithm with T. Since $\mathbf{A}_t$ is independent of the service process in any case, and doesn't depend on T, it suffices to show that $\mathrm{Var}\{\sum_{k=0}^{t-1}\sigma(k,T)\}$ is non-increasing in T for any given $t > 0$.

Define $r(k,T) = \mathrm{Cov}\{\sigma(0,T),\sigma(k,T)\}$ to be the covariance function in the steady state. We then have

$$\mathrm{Var}\left\{\sum_{k=0}^{t-1}\sigma(k,T)\right\} = t\cdot\mathrm{Var}\{\sigma(0,T)\} + \sum_{k=1}^{t-1}(t-k)r(k,T).$$

Since the marginal distribution remains the same under our algorithm, $\mathrm{Var}\{\sigma(0,T)\}$ does not depend on T. Thus, it remains to show that the second term on the RHS is non-increasing in T. Notice from Propositions 2 and 3, and by setting $f(\cdot)=g(\cdot)=\mathbf{1}_{\{\in B_v\}}$ that

$$r(k,T) = 0, \quad \text{for } k \neq nT,\ n = 1,2,\ldots \tag{10}$$
$$r(nT,T) = r(n,1) \triangleq r(n), \tag{11}$$

under our algorithm with parameter T. Observe that

$$\begin{aligned}
&\sum_{k=1}^{t-1}(t-k)r(k,T) \overset{(10)}{=} \sum_{k=T,2T,\ldots}^{t-1}(t-k)r(k,T)\\
&= \sum_{j=1}^{\lfloor (t-1)/T\rfloor}(t-jT)r(jT,T) \quad \text{(by letting } k=jT)\\
&\overset{(11)}{=} \sum_{j=1}^{\lfloor (t-1)/T\rfloor}(t-jT)r(j) \geq \sum_{j=1}^{\lfloor (t-1)/(T+1)\rfloor}(t-j(T+1))r(j)\\
&\overset{(11)}{=} \sum_{j=1}^{\lfloor (t-1)/(T+1)\rfloor}(t-j(T+1))r(j(T+1),(T+1))\\
&= \sum_{k=T+1,2(T+1),\ldots}^{t-1}(t-k)r(k,(T+1)) \overset{(10)}{=} \sum_{k=1}^{t-1}(t-k)r(k,(T+1))
\end{aligned}$$

where the inequality is due to $r(k) = \psi(k)\mathrm{Var}\{\sigma(k,1)\} \geq 0$ from the assumption. This completes the proof. ∎

Note that our algorithm with the order parameter T does not make an improvement upon every lag of correlation. As can be seen from the Figure 3(b), for example, $\psi(5,5)$ is larger than $\psi(5,1)$, so correlation can be even worse at some time lags. However, the Proposition 4 states that the *variance* of cumulative service process for any time scale t decreases monotonically, as T increases.

We expect that positive temporal correlations in the link service process under the original Glauber dynamics prevail for most cases of network topologies and are in accordance with wide-spread link starvation problem in CSMA scheduling. Recall that we have shown in Proposition 1 that $\psi(k)$ is lower bounded by a well defined positive function for every even k. If we consider a chain for the standard Glauber dynamics where the correlations are negative for odd-lags, the reduction in variance may not be guaranteed; However, as discussed in section III-B, the impact of correlations at even lags will be higher than those at odd lags, and therefore our high-order chain approach will benefit from the zero-padding and lag-shifting by outweighing the potential loss in the improvement.

B. Impact on transient behavior

Although the two properties, zero-padding and lag-shifting, of our algorithm are well understood in the steady state, they do not capture the behavior in the transient states. To understand such transient behavior, we utilize a notion of *mixing time* of a Markov chain P with its stationary distribution π, defined as follows [36], [35], [43]

$$t_{\mathrm{mix}}(\epsilon) \triangleq \min\{t : \max_{x\in\Omega}\|P^{(n)}(x,\cdot) - \pi\|_{\mathrm{TV}} \leq \epsilon,\ \ \forall n \geq t\},$$

where $P^{(n)}(x,\cdot)$ is n-step transition probability distribution starting from x. For general, non-Markov processes with the same stationary distribution π on the same state space, the mixing time can be similarly defined by considering $\|\mu_t - \pi\|_{\mathrm{TV}}$, where μ_t is the distribution at time t and maximizing over all possible initial distribution μ_0. Clearly, our delayed CSMA algorithm with order parameter T yields $\mu_t = \mu_{t-T}P$ when the transition kernel is time-homogeneous. Iteratively applying such relation gives $\mu_{nT} = \mu_0 P^{(n)}$, implying that in terms of the mixing time we need roughly T times larger number of state transitions to achieve the same degree of accuracy in distributional error, compared to the conventional CSMA-based algorithm. In other words, in our high-order Markov chain based approach, while the stationary distribution of the schedule itself remains untouched and it provides better queueing performance in the steady state, it may take roughly T times longer to reach the steady state.[2] This tradeoff between a bit slower convergence but to a 'better' stationary regime, also implies that the performance during the transient phase can be worse than that of the conventional CSMA algorithm (albeit our algorithm will eventually pay off in the end), and necessitates a careful choice of T depending upon how long the system is meant to be running.

As a remedy for such a problem, we here provide a gentler start-up algorithm, which combines the strengthes of relatively faster mixing of the traditional CSMA, and the reduced correlations of our delayed CSMA. This method consists of two steps. First, we run the conventional CSMA algorithm with $T = 1$ (so that the mixing time doesn't get hurt) until it gets reasonably closer to its stationary regime, and then *samples* T schedules with inter-spacing M between two consecutive samples, which will then be used as initial states to run our delayed CSMA algorithm. Specifically, after running the traditional CSMA for a while, at a certain time instant agreed among links ($t = 0$), each link keeps its channel state information in its memory at every M time slots. It will have T sampled channel states stored after MT time slots. Then, each link performs the delayed CSMA with the sampled states as the first T initial states. This procedure is illustrated in Figure 5.

[2]This also suggests that the mixing time based delay analysis may give looser bound on the average queue length *in the steady-state*. See [20] for similar accounts on the usage of mixing time for delay analysis.

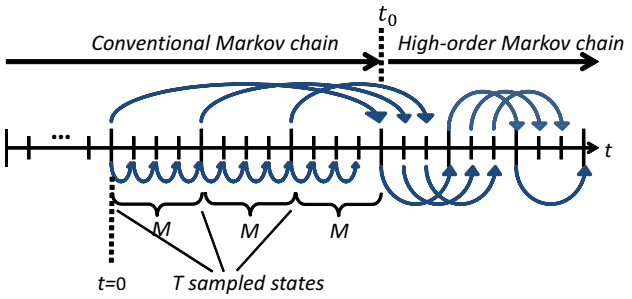


Fig. 5. An illustration of the gentler start-up algorithm. Blue arrows indicate state transitions. The algorithm starts at t_0 with T initially sampled states.

To better understand the effect of M, consider a configuration state X_t at time t, and suppose the conventional CSMA has run sufficiently long enough so that the marginal distribution can be assumed to be π, i.e., $\mathbb{P}\{X_t = x\} = \pi(x)$ for all $t \geq 0$. Without loss of generality, we index time $t = 0$ at which the sampling starts. To proceed, we define the 'self-adjoint' property of the operator $\mathbf{P}$ for a reversible chain with respect to π [36], [35], [43].

$$\begin{aligned}\langle f, \mathbf{P}g\rangle_\pi &= \sum_{x,y\in\Omega} f(x)P(x,y)g(y)\pi(x) \\ &= \sum_{x,y\in\Omega} f(x)P(y,x)g(y)\pi(y) = \langle \mathbf{P}f, g\rangle_\pi\end{aligned}$$

where the second equality is from the reversibility. Iteratively applying the above relation gives

$$\langle \mathbf{P}^{(n)}f, \mathbf{P}^{(m)}g\rangle_\pi = \langle \mathbf{P}^{(n+m)}f, g\rangle_\pi = \langle f, \mathbf{P}^{(n+m)}g\rangle_\pi, \quad (12)$$

for any $n, m \geq 0$.

Now, consider the correlation between X_{t_0+T} and X_{t_0+T+1} for example, where $t_0 = MT$ is the time that the delayed CSMA starts as in Figure 5. Note that

$$\mathbb{E}\{f(X_{t_0})g(X_{t_0+1})\} = \sum_{x\in\Omega} \mathbb{E}\{f(X_{t_0})g(X_{t_0+1})|X_0 = x\}\pi(x).$$

Conditioning on $X_0 = x$ gives two disjoint sample paths leading to X_{t_0+T} and X_{t_0+T+1} as follows:

$$\begin{aligned}&(i) X_0 = x \to X_{t_0} \to X_{t_0+T} \\ &(ii) X_0 = x \to X_1 \to X_2 \to \cdots \to X_M \to X_{t_0+1} \to X_{t_0+T+1}.\end{aligned}$$

Thus, by the conditional independence, we have

$$\begin{aligned}&\mathbb{E}\{f(X_{t_0+T})g(X_{t_0+T+1}) \mid X_0 = x\} \\ &= \mathbb{E}\{f(X_{t_0+T})|X_0 = x\}\mathbb{E}\{g(X_{t_0+T+1})|X_0 = x\} \\ &= (\mathbf{P}^{(2)}f)(x)(\mathbf{P}^{(M+2)}g)(x),\end{aligned}$$

and therefore

$$\begin{aligned}&\mathbb{E}\{f(X_{t_0+T})g(X_{t_0+T+1})\} \\ &= \sum_{x\in\Omega} (\mathbf{P}^{(2)}f)(x)(\mathbf{P}^{(M+2)}g)(x)\pi(x) \\ &= \langle \mathbf{P}^{(2)}f, \mathbf{P}^{(M+2)}g\rangle_\pi = \langle f, \mathbf{P}^{(M+4)}g\rangle_\pi,\end{aligned}$$

where the last equality is from (12). This implies that although the time difference between X_{t_0+T} and X_{t_0+T+1} is only one, its correlation behavior is of lag $M+4$.

For general cases of two points $t_0 + i$ and $t_0 + j$ with $t_0 = MT$ as in Figure 5, by counting the number of transitions from X_0 as we did in the above, we arrive to the following.

Proposition 5. *Let $t_0 + i = nT + m$, $t_0 + j = n'T + m'$ for $m, m' = 0, \ldots, T-1$, and $n, n' = 1, 2, \ldots$. Then,*

$$\mathbb{E}\{f(X_{t_0+i})g(X_{t_0+j})\} = \langle f, \mathbf{P}^{(k)}g\rangle_\pi,$$

where

$$k = \begin{cases} |n - n'|, & \text{if } m = m', \\ n + n' + |m - m'|M + 2 & \text{otherwise.} \end{cases}$$

Proposition 5 implies zero-padding and lag-shifting properties as shown in Propositions 2 and 3, by taking limits on i, j together and considering $i \not\equiv j \pmod T$ and $i \equiv j \pmod T$, respectively.[3] The choice of parameter M offers another tradeoff between quicker start-up and convergence speed. For instance, if M is very large so that the first T initial samples are almost independent of each other, then we have zero-padding correlation behavior from the beginning, but it would take very long to prepare such T independent samples. On the other hand, if M is small (say, $M = 1$), we have the opposite situation with slower convergence from a quicker start.

C. Throughput optimality under dynamic fugacity

So far, we have assumed that the fugacity parameters are fixed to be some constants, but in practice, it will be more desirable to adjust them dynamically in order to support required link arrival rates. Under some mild assumptions, the authors in [12], [13] proposed to use $\lambda_v(t) = e^{W_v(t)}$ where the weights $W_v(t)$ are in the form of $\log\log(Q_v(t))$, and have shown that the conventional CSMA algorithm via Glauber dynamics achieves the throughput optimality. And also, in [14], the choice of $\log\log(\cdot)$ for $h(\cdot)$ has been slightly generalized into $\log(\cdot)/g(\cdot)$ for a function g that increases arbitrarily slowly. We here verify that the throughput optimality can similarly hold for any parameter T in our algorithm.

Although the link schedules under our algorithm are updated based on their T-step-back states, we set the weight function $W_v(t)$ to be a function of the current queue length $Q_v(t)$ at time t, rather than T steps ago, such that the system can react more quickly by adjusting the fugacities with the latest information. With the time-varying fugacities, the state transition matrix becomes time-inhomogeneous, which we write as P_t, a function of $\lambda_v(t), v \in \mathcal{N}$ at time t. For each given such P_t, let π_t be its unique stationary distribution (in a row vector form), i.e., $\pi_t = \pi_t P_t$, and μ_t be the actual distribution of the link schedules $\boldsymbol{\sigma}(t)$ at time t under our algorithm with order parameter T. Then, we have

$$\mu_t = \mu_{t-T}P_{t-1}. \quad (13)$$

Similar to the steps via 'network adiabatic' theorem in [12], [13], a key step in proving the throughput optimality of our algorithm is to show that $\mu_t \approx \pi_t$ for sufficiently large queue lengths under (13). The distance between two

[3]This is because $\lim_{k\to\infty} r(k) = 0$ since $\mathbb{P}\{X_k \in B|X_0 \in B\} - \mathbb{P}\{X_0 \in B\} = \sum_{j=2}^{|\Omega|} \alpha_j \rho_j^k$ from (4) and $|\rho_j| < 1$ for $j, = 2, \ldots, |\Omega|$.

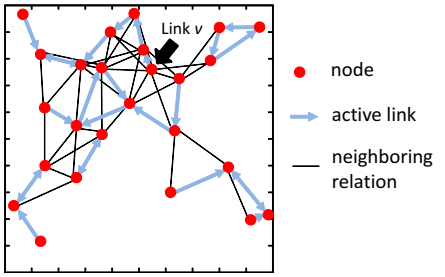


Fig. 6. An instance of network configuration with 25 nodes

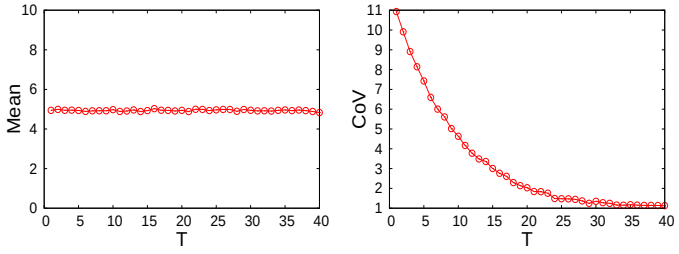
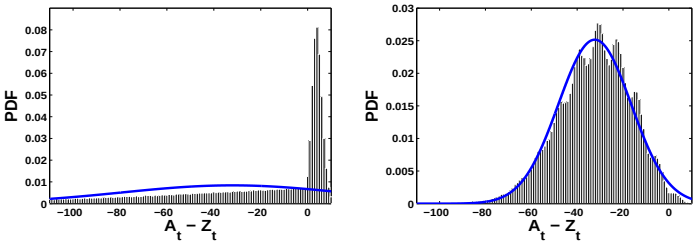


Fig. 7. Mean and CoV of 'off' duration for $\sigma_v(t)$



(a) T=1, t=217 (b) T=25, t=217

Fig. 8. Histogram of $\mathbf{A}_t - \mathbf{S}_t$ and its approximation by a Gaussian process. The lines are Gaussian distribution drawn from measured sample mean and variance.

probability distributions is characterized by the notion of total variation (TV) distance [36], [35], defined as $\|\nu - \mu\|_{\text{TV}} = \frac{1}{2}\sum_{\sigma\in\Omega}|\nu(\sigma) - \mu(\sigma)|$. Our approach toward the throughput optimality of our proposed algorithm is similar to those in [12], [14], [13], with a little modifications. First, suppose P_t is changing very slowly over time t for all large queue-lengths. (In fact, this can be achieved by setting the weight function $W_v(t)$ as a very slowly increasing function of the queue-length [12], [14], [13].) Then, the resulting π_t, a solution to $\pi = \pi P_t$, is also slowly varying over time t such that the actual distribution μ_t is able to get closer to π_t (in the sense of TV distance) before π_t moves away to another, thus effectively simulating the separation of timescales. As will be shown in Section IV-B, under our algorithm with order parameter T, the speed of convergence of μ_t under static fugacity (i.e., static P) is roughly T times slower than that of the conventional Glauber dynamics. This implies that the speed of actual distribution μ_t under dynamic fugacity as a slowly varying function of queue-length, will also be reduced by a factor of T. Since T is finite, we expect that μ_t is still able to catch up the slowly varying target π_t in time, and by following similar steps in [12], [14], [13] our algorithm can also achieve the throughput optimality, as shown in the following proposition. We provide its proof in the Appendix section.

Proposition 6. *Let $\epsilon > 0$ be arbitrarily given. For any arrival rate $\boldsymbol{\eta} \in (1-\epsilon)\mathbb{C}$, set the dynamic link weight* [4]

$$W_v(t) = \max\left\{h(Q_v(t)), \frac{\epsilon}{2|\mathcal{N}|}h(Q_{max}(t))\right\}, \qquad (14)$$

where $h(\cdot) = \log\log(\cdot + e)$. Then, our algorithm with any finite order parameter T satisfies (3), and is thus throughput optimal.

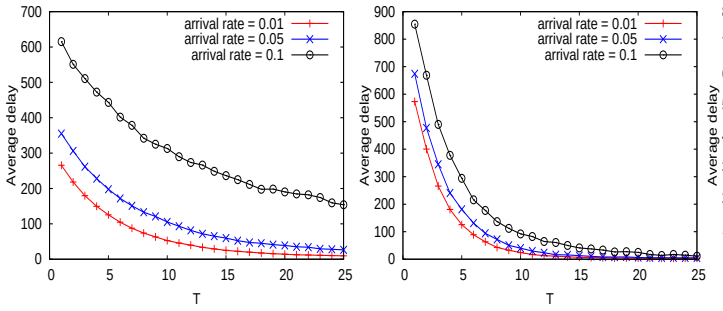
V. Simulation Results

In this section, we present simulation results for the delayed CSMA algorithm. We consider a network scenario by a random geometric graph (RGG) model with 25 nodes uniformly and independently positioned over the $1000 \times 1000m^2$ area. The transmission range of each node is set to be $250m$, where two nodes can communicate with each other if they are in the range of transmission. We have each node select one of its neighboring node uniformly at random, and create a communication link. If the receiver node of a link is within a range of transmitter node of another link, we consider that the two links form an edge in the conflict graph. If a link is scheduled for a time slot, a single packet in its queue is served when the queue is not empty. For the decision schedule, we choose the access probabilities $a_i = 0.25$ for links $i \in \mathcal{N}$. In this scenario, we collect simulation results for the link v shown in Figure 6. In obtaining simulation data, we have taken average of the data from 10000 time repeated simulations, and unless otherwise noted, for each of the runs we discard the data form the first half of the simulation time, to obtain the results in the steady-state.

First, to understand the benefits of having more drastic changes in the zero-one service process $\sigma_v(t)$ under our delayed CSMA algorithm, we look at the distribution of its 'off' duration U, the duration from an active slot to the next active slot. We measured its mean $\mathbb{E}\{U\}$ and the coefficient of variation (CoV) $\sqrt{\text{Var}\{U\}}/\mathbb{E}\{U\}$ as we increase the order parameter T, where we set $\lambda_i = 1$ for all $i \in \mathcal{N}$. Figure 7 shows that the first-order statistics of the link state doesn't change with T, while its variability decreases for larger T, implying that our algorithm with larger T effectively removes link starvation phenomenon.

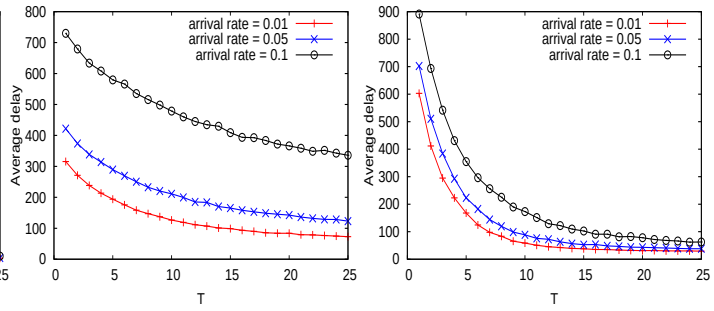
Before investigating the delay performance of our proposed algorithm, we first validate our approximation. As before, under the static fugacity setup, we first collect the cumulative net-input $\mathbf{A}_t - \mathbf{S}_t$ up to $t = 217$ from a stationary start $t = 0$ where the arrival rates are all set to be $\eta_v = 0.1$. As seen in Figure 8(a), the net-input process in the standard CSMA case ($T = 1$) is far from being Gaussian due to the strong correlation structure in $\sigma_v(t)$. As we increase the order parameter to $T = 25$, it becomes very close to Gaussian as

[4] $Q_{\max}(t) = \max_{i\in\mathcal{N}} Q_i(t)$. In [12] it is argued that $Q_{\max}(t)$ can be easily estimated through a gossip-like message passing mechanism. For simplicity, we assume $Q_{\max}(t)$ is known throughout, and do not discuss this issue here.



(a) $\lambda = 1$ (b) $\lambda = 5$

Fig. 9. Delay performance of the delayed CSMA with different order parameters T where the packet arrival process is Bernoulli.



(a) $\lambda = 1$ (b) $\lambda = 5$

Fig. 10. Delay performance of the delayed CSMA with order parameter T where the packet arrival process is based on a two-state, *arrival* and *idle*, Markov chain.

shown in Figure 8(b). The same conclusion also applies to the modified net-input $\mathbf{Z}(t) = \mathbf{A}_t - \mathbf{S}_t + Ct$.

We evaluate the delay performance of our algorithm with appropriately chosen arrival rates and fugacity parameters. We consider two cases of $\lambda_i = 1$ or 5 for all $i \in \mathcal{N}$, where the average packet arrival rate for the link v is chosen to be $0.01, 0.05$ and 0.1, respectively. The Figure 9 shows the delay performance measured at the link v where the packet arrival process follows the Bernoulli process, i.e., $\mathbb{E}\{A_v(t) = 1\} = \eta_v$ where $\eta_v = 0.01, 0.05$ and 0.1. The results show that our algorithm achieves smaller average delay as we use larger order parameter T.

We also consider the cases where the packet arrival process is not Bernoulli process but based on a Markov chain with its transition probability, $\mathbb{P}\{A_v(t) = 1|A_v(t) = 0\} = 1 - \mathbb{P}\{A_v(t) = 0|A_v(t) = 0\} = p$, and $\mathbb{P}\{A_v(t) = 0|A_v(t) = 1\} = 1 - \mathbb{P}\{A_v(t) = 1|A_v(t) = 1\} = q$. The degree of variability of this process is captured by the second largest eigenvalue of this Markov chain, which is determined by $1 - p - q \triangleq \delta$. When $\delta = 0$, the process is identical to that of Beroulli process, whereas we used $\delta = 0.9$ for our simulations, which generates more bursty packet arrivals. In order to focus on the impact of the arrival pattern, we fixed the long term average arrival rate, $\lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} \mathbb{P}\{A_v(t) = 1\} = \frac{p}{p+q} = \bar{\eta}_v$ where $\bar{\eta}_v = 0.01, 0.05$ and 0.1. For given $\bar{\eta}_v$ and δ, one can find the transition probability p and q corresponding to those $\bar{\eta}_v$ and δ, and hence we changes p and q accordingly. The Figure 10 shows that our algorithm improves the delay performance in the cases where the arrival process is not Bernoulli as well.

In addition, we run simulations with dynamic fugacity scheme where the weight function is set to be $\log\log(Q_i(t) + e)$ for all $i \in \mathcal{N}$. We have measured the delay of each packet in the queue until gets served, under different traffic intensity. We adjusted arrival rate of each link by gradually increasing the rate proportional to its potential link capacity, where the link capacity is calculated by summing over all possible maximal independent sets with equal weight. As the traffic intensity increases to 1, the rate approaches to maximum throughput. The left-hand side of Figure 11 shows the delay improvement over the conventional CSMA algorithm, where the inset figure displays the ratio of the delay under our algorithm with chosen T to that of standard CSMA. We note that the performance

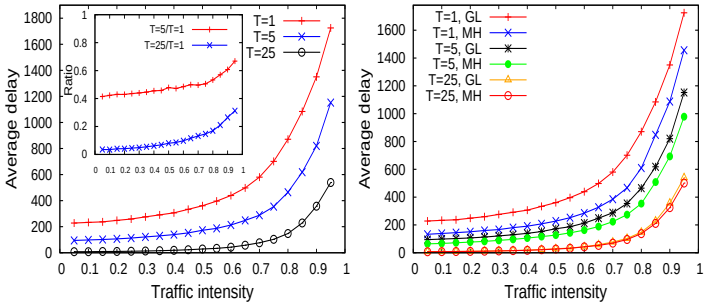
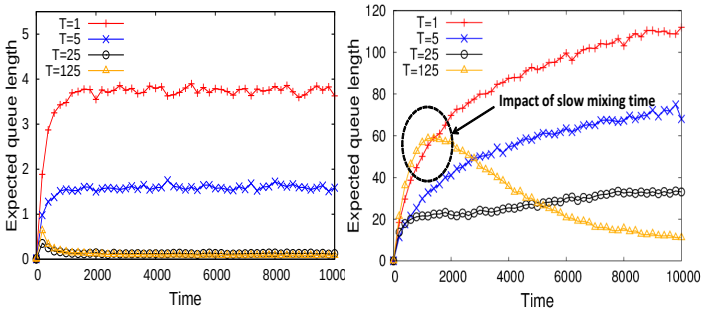


Fig. 11. Delay performance of the delayed CSMA under dynamic fugacity while varying traffic intensity.

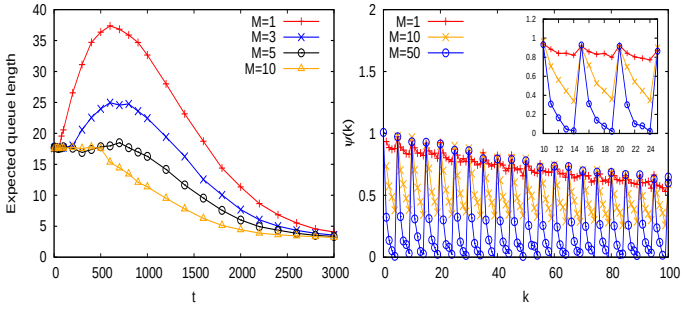
improvement is quite remarkable. For instance, with $T = 5$, the delay reduces by half, and with $T = 25$, it reduces by a factor of 20 compared to the conventional CSMA algorithm. We observed similar trends in the improvement under different weigh functions such as $\log(Q_i(t) + 1)$ or $Q_i(t)$. There is a tendency in the improvement ratio to increases (yet smaller than 1) as the traffic intensity approaches to the edge of the capacity region. This is largely because in heavy traffic regime, the average service rate is more dominant factor of the queueing delay than its variability, whereas our algorithm improves only on the variance.

Recall that our approach is to add 'order of T' into the original Glauber dynamics to keep the same marginal distribution while shaping the correlation structure to our advantage. We note that this approach will hold for any other standard algorithm modeled by a reversible Markov chain on the same feasible state space achieving the same stationary distribution, not necessarily the Glauber dynamics considered so far. For example, it was shown in [19] that Metropolis-Hastings (MH) based algorithm (still reversible Markov chain on Ω) outperforms the conventional Glauber dynamics. The right-hand side of Figure 11 shows that indeed MH algorithm gives better delay performance than the standard Glauber dynamics (GL), but still, the amount of delay reduction is significant for larger T. Interestingly, the performance gap between the standard Glauber and the MH algorithm (all with $T = 1$) seems to narrow down for larger T, implying that the effect of correlation reductions via high-order Markov chain dominates over the choice of base reversible Markov chains.



(a) Traffic intensity=0.1 (b) Traffic intensity=0.9

Fig. 12. Impact of T on a queue evolution.



(a) $\mathbb{E}\{Q(t)\}$ with $T = 50$ (b) $\psi(k)$ with $T = 5$

Fig. 13. Impact of M on (a) queue-length evolution and (b) correlation reduction

We look at the impact on transient behavior induced by our algorithm with different T. To see this, we have measured the queue-length from after simulation starts. In order to measure the queue-length during the initial phase, we collect queue-length samples and obtain its time average for different time indexes. Figure 12 shows the measured queue-length under different T. As discussed in section IV-B, we might expect larger backlog at initial phase when large T is used (due to slower mixing time). However, in practice, its impact is not significant unless indefinitely large T is used. For example, as can be seen from Figure 12(a), its impact is almost negligible when traffic intensity is low. Even under high traffic intensity (Figure 12(b)) and with large T, e.g., $T = 125$, the accumulation of backlog at transient phase due to slower mixing is still comparable to that of the original case ($T = 1$), and the queue eventually converges to smaller size in the steady state. Thus clearly, the benefit in long-term behavior outweighs such a drawback.

We also look at the benefit of utilizing another parameter M; the gentler start scheme proposed in section IV-B. In this case, we have run simulations under static fugacity with $\lambda_i = 1$ for all links $i \in \mathcal{N}$ in order to well observe the de-correlation process induced by the parameter M. Firstly, we have run the conventional CSMA algorithm for sufficiently long enough time, and then measured the queue length right after the delayed CSMA algorithm starts to run. Figure 13(a) shows the measured queue-length with $T = 50$ under different M. For $M = 1$ (quick start), large order parameter $T = 50$ leads to worse performance than the standard CSMA at the initial phase, but its strong correlation reduction property in the end eventually offsets such a drawback. As discussed in SectionIV-B, such an effect in the transient phase can be alleviated by sampling intermediate states that are separated by M. We can see that fairly small values of M can readily reduce the drawback. Larger M leads to quicker correlation reduction as seen in Figure 13(b) (for lags not a multiple of $T = 5$ here), and also as predicted from Proposition 5.

VI. Conclusion

In this paper, we have proposed the delayed CSMA algorithm based on a high-order Markov chain on the same state space of feasible schedules achieving the same stationary distribution, but with different and reshaped correlation structure. Our algorithm is extremely simple to implement without requiring any additional message passing or overhead, with the exception of a single parameter T across the network once and for all. We have proved that our algorithm is throughput-optimal, yet provides far better delay performance by emulating the effect of superposition of independent traffic streams in the queue, out of a single input and still single service process, via zero-padding and lag-shifting properties of the resulting service process of the queue under our algorithm.

Another interesting viewpoint arising from our investigation is that the conventional approach via mixing time for delay performance must be used with great care, since in our setting we achieve better queueing performance in the steady-state by trading much less correlations for a bit slower mixing speed. While faster mixing and less correlations typically come in pair under the traditional Markov chain based approach, we note that our high-order Markov chain (or simply non-Markov on the same state space) can now separate these two, thus enabling us to trade one for the other, toward better understanding and design of distributed schedulers running over different timescales of interest.

Appendix

Proof of Proposition 6: First, we need the following.

Lemma 2. *[23] For a scheduling algorithm, given any $0 < \epsilon, \delta < 1$ and if there exists $0 < B(\delta, \epsilon) < \infty$ such that in any time slot t, with probability larger than $1 - \delta$, the scheduling algorithm chooses a schedule $\sigma(t) \in \Omega$ that satisfies*

$$\sum_{v \in \boldsymbol{\sigma}(t)} W_v(t) \geq (1 - \epsilon) \max_{\boldsymbol{\sigma} \in \Omega} \sum_{v \in \boldsymbol{\sigma}} W_v(t) \tag{15}$$

whenever $\max_{v \in \mathcal{N}} Q_v(t) > B(\delta, \epsilon)$, then the scheduling algorithm is throughput-optimal in the sense of (3).

We basically extend the procedure used for proving the throughput optimality for conventional CSMA algorithm in [14]. The key steps therein can be summarized as follows. (See [14] for details.)

1) Given $\delta \in (0, 1)$, find a $K(\delta)$ such that

$$\|\pi_t - \mu_t\|_{TV} \leq \frac{\delta}{4} \tag{16}$$

holds whenever $Q_{\max}(t) \geq K(\delta)$.

2) Given $\epsilon \in (0,1)$, choose $B(\delta,\epsilon)$ as
$$B(\delta,\epsilon) = \max\left\{K(\delta), h^{-1}\left(\frac{|\mathcal{N}|\log 2 + \log\frac{2}{\delta}}{\epsilon/2}\right)\right\}.$$
3) Then for any arrival rate $\boldsymbol{\eta} \in (1-\epsilon)\mathbb{C}$, Lemma 2 can be established and thus the scheduling algorithm is throughput optimal.

Since the delayed CSMA algorithm yields $\mu_t = \mu_{t-T}P_{t-1}$ as opposed to $\mu_t = \mu_{t-1}P_{t-1}$ in the standard CSMA algorithm, we will have to prove the step 1 above in our setting, and finding a $K(\delta)$ under our algorithm with order parameter T is the key challenge in proving throughput optimality. This statement is established in the following.

Lemma 3. *Given any $\delta \in (0,1)$ and under our delayed CSMA with parameter T, there exists $K(\delta) < \infty$ such that $\|\mu_t(\sigma) - \pi_t(\sigma)\|_{TV} \le \frac{\delta}{4}$ holds whenever $Q_{\max}(t) \ge K(\delta)$.*

The rest of this paper is devoted to the proof of Lemma 3 and this will complete our proof of Proposition 6. To proceed, we collect some notations and useful lemmas first.

Definition 1. *[36] (χ^2 distance) The χ^2 distance between two probability distributions ν and μ on a finite space Ω is defined by*
$$\left\|\nu-\mu\right\|^2_{\frac{1}{\mu}} = \sum_{\sigma\in\Omega}\frac{1}{\mu(\sigma)}\left(\nu(\sigma)-\mu(\sigma)\right)^2.$$

Then, the following relationship holds [36].
$$\left\|\nu-\mu\right\|_{\frac{1}{\mu}} = \left\|\frac{\nu}{\mu}-1\right\|_{\mu} \ge 2\|\nu-\mu\|_{TV}. \tag{17}$$

Lemma 4. *(*The $1/\pi$-bound*) [36] Let P be an aperiodic, irreducible, and reversible transition matrix on the finite space Ω, with its stationary distribution π. Let $1=\rho_1>\rho_2\ge\cdots\ge\rho_{|\Omega|}>-1$ be the eigenvalues of P. Then for any probability distribution ν on Ω, and for all $n\ge 1$,*
$$\left\|\nu P^n - \pi\right\|_{\frac{1}{\pi}} \le \rho(P)^n\left\|\nu-\pi\right\|_{\frac{1}{\pi}} \tag{18}$$
where $\rho(P) = \max\{\rho_2, |\rho_{|\Omega|}|\}$ is the second largest eigenvalue modulus (SLEM) of P.

Lemma 5. *Given $t,k\in\mathbb{N}$, define*
$$\alpha_t = \sum_{i\in\mathcal{N}}\left\{h'(\hat{Q}_v(t)) + h'(\hat{Q}_v(t+1))\right\}$$
where $\hat{Q}_v(t) = h^{-1}(W_v(t))$. If $\alpha_t < \frac{1}{k}$, then
1) $1-k\alpha_t \le \frac{\pi_{t+k}(\sigma)}{\pi_t(\sigma)} \le 1+k\alpha_t$, $\forall\sigma\in\Omega$.
2) $\left\|\pi_{t+k}-\pi_t\right\|_{\frac{1}{\pi_{t+k}}} \le 2k\alpha_t$.

The above is a generalized version of Lemma 13 in [12], and is straightforward to verify, so we omit the proof here.

In determining the decision schedule $m(t)$ at time t, without loss of generality, we assume a single site update rule, i.e., only one link is selected uniformly at random at each time slot. Our analysis can then be extended to the case of multiple site updates by following the same steps of Lemma 7 in [14]. Thus, we will mainly refer to the analysis for single site update as given in Lemma 3 of [14], which is reproduced below. From this point on, for notational simplicity, we set N as the number of links in the conflict graph, i.e., $N = |\mathcal{N}|$.

Lemma 6. *Let $M_t = \frac{1}{1-\rho(P_t)}$, where $\rho(P_t)$ is the SLEM of transition probability matrix P_t. Then, we have*
$$M_t \le 16^N e^{(4NW_{\max}(t))} \tag{19}$$
where $W_{\max}(t) = \max_{v\in\mathcal{N}} W_v(t)$

We provide the following lemma that is essential to our proof, which is a modification of Lemma 14 in [12], but with our order parameter T in mind.

Lemma 7. *Given any $\delta\in(0,1)$, define a constant $B = B(N,\delta)$ satisfying $B \ge (16N-1)^{16N-1}$ and*
$$\frac{64T16^N N\log^{4N}(x+T-1+e)}{\exp\left((\log(x+e))^{\frac{\delta}{2N}}\right)-1-e} < \delta \tag{20}$$
for all $x\ge B$. If $Q_{\max}(t)\ge B$, then
$$M_{t+T-1}\cdot\alpha_t \le \frac{\delta}{32T}, \quad \textit{and} \quad M_{t+T-1}\cdot\alpha_{t-1} \le \frac{\delta}{32T}.$$

Proof: Note that $\hat{Q}_{\min} = h^{-1}(W_{\min}) \ge h^{-1}\left(\frac{\delta}{2N}h(Q_{\max})\right)$ from (14) and h^{-1} is increasing. Using $h'(x) = \frac{1}{(x+e)\log(x+e)} < \frac{1}{x}$ for $x>0$, and the fact that $Q(t)-k \le Q(t+k) \le Q(t)+k$ for any $k\ge 1$, we have, for $i=0,1$,
$$\alpha_{t-i} \le \frac{N}{\hat{Q}_{min}(t-i)} + \frac{N}{\hat{Q}_{min}(t+1-i)} \le \frac{2N}{\hat{Q}_{min}(t)-1}$$
Also, from (19), $M_{t+T-1} \le 16^N\log^{4N}(\hat{Q}_{\max}(t)+T-1+e)$. Then, for $i=0,1$,
$$\begin{aligned} M_{t+T-1}\cdot\alpha_{t-i} &\le \frac{2N16^N\log^{4n}\left(\hat{Q}_{\max}(t)+T-1+e\right)}{\hat{Q}_{min}(t)-1} \\ &\le \frac{2N16^N\log^{4n}(x+T-1+e)}{e^{(\log(x+e))^{\frac{\delta}{2N}}}-1-e} \end{aligned} \tag{21}$$
where $x := \hat{Q}_{\max}(t) \ge B$. From (20) and since the term in (21) goes to zero as $x\to\infty$, it is bounded above by $\frac{\delta}{32T}$. ■

The following is the main statement in proving the throughput optimality, which is also similarly used in [12], [14].

Lemma 8. *At time t, if $Q_{\max}(t)\ge B+t^*$ where t^* is*
$$T^2\left\lceil 16^{2N}\log^{8N}(B+T+e)\log\left(\frac{4}{\delta}(2\log(B+T-1+e))^{N/2}\right)\right\rceil^2$$
Then,
$$\left\|\mu_t-\pi_t\right\|_{\frac{1}{\pi_t}} \le \delta/2. \tag{22}$$

Proof: We follow similar steps as in the proof of Lemma 12 in [12] with some modifications. First, by applying triangle inequality, we have
$$\begin{aligned} \left\|\mu_t-\pi_t\right\|_{\frac{1}{\pi_t}} &\le \left\|\mu_t-\pi_{t-1}\right\|_{\frac{1}{\pi_t}} + \left\|\pi_{t-1}-\pi_t\right\|_{\frac{1}{\pi_t}} \\ &\le (1+\alpha_{t-1})\left\|\mu_t-\pi_{t-1}\right\|_{\frac{1}{\pi_{t-1}}} + 2\alpha_{t-1}, \end{aligned}$$
where the second inequality is from Lemma 5, and the fact

that $\alpha_{t-1} < \frac{\delta}{32} < 1$. We can see that (22) holds if

$$\left\|\mu_t - \pi_{t-1}\right\|_{\frac{1}{\pi_{t-1}}} \le \delta/4.$$

Define $r_j = \|\mu_j - \pi_{j-1}\|_{\frac{1}{\pi_{j-1}}}$. Using the Lemma (18) and triangle inequality again, we have

$$\begin{aligned} r_{j+T} &= \left\|\mu_{j+T} - \pi_{j+T-1}\right\|_{\frac{1}{\pi_{j+T-1}}} \\ &= \left\|\mu_j P_{j+T-1} - \pi_{j+T-1}\right\|_{\frac{1}{\pi_{j+T-1}}} \quad \text{(from (13))} \\ &\le \rho(P_{j+T-1})\left\|\mu_j - \pi_{j+T-1}\right\|_{\frac{1}{\pi_{j+T-1}}} \\ &\le \rho(P_{j+T-1})\left(\|\mu_j - \pi_j\|_{\frac{1}{\pi_{j+T-1}}} + \|\pi_j - \pi_{j+T-1}\|_{\frac{1}{\pi_{j+T-1}}}\right). \end{aligned} \tag{23}$$

Let t_0 be the latest time that $Q_{\max}$ hits B, i.e., $t_0 = t - \Delta t_0$, where

$$\Delta t_0 = \min\{\Delta t \ge 0 : Q_{\max}(t - \Delta t) = B\}.$$

Since $Q_{\max}(j) \ge B$ for all $t_0 \le j \le t$, we have for such j,

$$\begin{aligned} &\|\mu_j - \pi_j\|_{\frac{1}{\pi_{j+T-1}}} \\ &\le (1 + (T-1)\alpha_j)\|\mu_j - \pi_j\|_{\frac{1}{\pi_j}} \\ &\le (1 + (T-1)\alpha_j)((1+\alpha_{j-1})r_j + 2\alpha_{j-1}) \\ &\le \left(1 + \frac{\delta}{32M_{j+T-1}}\right)\left(\left(1 + \frac{\delta}{32M_{j+T-1}}\right)r_j + \frac{\delta}{16M_{j+T-1}}\right) \end{aligned} \tag{24}$$

where the last inequality is from $(T-1)\alpha_j \le \frac{(T-1)\delta}{32TM_{j+T-1}} \le \frac{\delta}{32M_{j+T-1}}$. Similarly,

$$\left\|\pi_j - \pi_{j+T-1}\right\|_{\frac{1}{\pi_{j+T-1}}} \le 2(T-1)\alpha_j \le \frac{\delta}{16M_{j+T-1}}. \tag{25}$$

If we assume $r_j < \delta/4$, then it can be checked that (24) + (25) $< \frac{\delta}{4} + \frac{\delta}{4M_{j+T-1}}$, and hence from (23), (24), and (25),

$$\begin{aligned} r_{j+T} &< \rho(P_{j+T-1})\left(\frac{\delta}{4} + \frac{\delta}{4M_{j+T-1}}\right) \\ &= \left(1 - \frac{1}{M_{j+T-1}}\right)\left(\frac{\delta}{4} + \frac{\delta}{4M_{j+T-1}}\right) \le \frac{\delta}{4}. \end{aligned}$$

Therefore, if there exists some $\tilde{k}$ that satisfies the conditions **(C1)** $r_{t-\tilde{k}T} \le \delta/4$, and **(C2)** $\tilde{k} \le \frac{t-t_0}{T}$, then we will have $r_t \le \delta/4$.

Let k' be the largest k such that $t_0 \le t - kT$, and let $t_0' = t - k'T$. Then it can be verified that finding $\tilde{k}$ is equivalent to finding k^* such that **(C1')** $r_{t_0'+k^*T} \le \delta/4$, and **(C2')** $k^* \le \frac{t-t_0}{T}$. To find k^*, assume $r_{t_0'+kT} > \delta/4$ for all $k < k^*$, then we get the following upper bound for $j = t_0'+T, t_0'+2T, \ldots, t_0'+k^*T$.

$$\begin{aligned} r_j &\le \left(1 - \frac{1}{M_{j-1}}\right)\left(\left(1 + \frac{\delta}{32M_{j-1}}\right)^2 r_{j-T} + \frac{\delta}{16M_{j-1}}\left(1 + \frac{\delta}{32M_{j-1}}\right)\right) \\ &\le \left(1 - \frac{1}{M_{j-1}}\right)\left(\left(1 + \frac{\delta}{32M_{j-1}}\right)^2 r_{j-T} + \frac{r_{j-T}}{8M_{j-1}}\left(1 + \frac{\delta}{32M_{j-1}}\right)\right) \\ &\le \left(1 - \frac{1}{M_{j-1}}\right)\left(1 + \frac{1}{M_{j-1}}\right) r_{j-T} \\ &= \left(1 - \frac{1}{M_{j-1}^2}\right) r_{j-T} \;\le\; e^{-\frac{1}{M_{j-1}^2}} \cdot r_{j-T}. \end{aligned}$$

Then, we have $r_{t_0'+k^*T} \le r_{t_0'} \exp\left(-\sum_{j=1}^{k^*} \frac{1}{M_{t_0'+jT}^2}\right)$, where

$$\begin{aligned} \sum_{j=1}^{k^*} \frac{1}{M_{t_0'+jT}^2} &\ge \sum_{j=1}^{k^*} \frac{1}{16^{2N} e^{8Nf(Q_{\max}(t_0'+jT))}} \\ &= \sum_{j=1}^{k^*} \frac{1}{16^{2N}(\log(Q_{\max}(t_0'+jT)+e))^{8N}} \\ &\ge \sum_{j=1}^{k^*} \frac{1}{16^{2N}(\log(Q_{\max}(t_0')+k^*T+e))^{8N}} \\ &= \frac{k^*}{16^{2N}(\log(Q_{\max}(t_0')+k^*T+e))^{8N}} \\ &\ge \frac{\sqrt{k^*}}{16^{2N}(\log(Q_{\max}(t_0')+1+e))^{8N}\sqrt{T}} \\ &\ge \frac{\sqrt{k^*}}{16^{2N}(\log(Q_{\max}(t_0)+T+e))^{8N}\sqrt{T}}. \end{aligned}$$

The third inequality is from the fact that $k^* \ge 1$, $Q_{\max}(t_0) = B \ge (16N-1)^{16N-1}$ and the following inequality in [12],

$$\sqrt{x} \ge \left(\frac{\log(x+y)}{\log(1+y)}\right)^{8N}, \quad \forall x \ge 1, y \ge (16N-1)^{16N-1}.$$

In addition,

$$\begin{aligned} r_{t_0'} &= \left\|\mu_{t_0'+1} - \pi_{t_0'}\right\|_{\frac{1}{\pi_{t_0'}}} \le \sqrt{\frac{1}{\min_{\boldsymbol{\sigma}\in\Omega}\pi_{t_0'}(\boldsymbol{\sigma})}} \le \sqrt{Z(t_0')} \\ &\le (2e^{h(Q_{\max}(t_0'))})^{N/2} = (2\log(Q_{\max}(t_0)+T-1+e))^{N/2}, \end{aligned}$$

where $Z(t)$ is the normalizing constant for the distribution π_t, i.e., $Z(t) = \sum_{\boldsymbol{\sigma}\in\Omega}\prod_{v\in\mathcal{N}}(\lambda_v(t))^{\sigma_v}$ with $\lambda_v(t) = e^{W_v(t)}$. If we choose k^* as,

$$T\left\lceil 16^{2N}\log^{8N}(B+T+e)\log\left(\frac{4}{\delta}(2\log(B+T-1+e))^{N/2}\right)\right\rceil^2$$

then it can be checked that $r_{t_0'+k^*T} \le \delta/4$, so **(C1')** is satisfied. And since $Q_{\max}(t_0) = B$ and $Q_{\max}(t) \ge B + t^*$ (note $t^* = k^*T$) as given in the lemma, we can check $t \ge t_0 + t^*$, which implies **(C2')**. This completes the proof of Lemma 8. ∎

Therefore, the proof of Lemma 3 completes by the choice of $K(\delta) = B(N,\delta) + t^*$, and from the inequality in (17).

REFERENCES

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, Dec. 1992.

[2] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *IEEE INFOCOM*, April 2008.

[3] X. Wu, R. Srikant, and J. R. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," in *IEEE INFOCOM*, May 2006.

[4] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest queue first scheduling: Second order properties using fluid limits," *Adv. Appl. Probab.*, vol. 38, no. 2, pp. 505–521, 2006.

[5] G. Zussman, A. Brzezinski, and E. Modiano, "Multihop local pooling for distributed throughput maximization in wireless networks," in *IEEE INFOCOM*, April 2008.

[6] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," in *ACM MobiHoc*, May 2009.

[7] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks a partitioning approach," in *ACM MobiCom*, September 2006.
[8] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *ACM SIGMETRICS/Performance*, June 2006.
[9] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *ACM Sigmetrics*, June 2007.
[10] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. on Networking*, vol. 18, no. 3, pp. 960–972, June 2010.
[11] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel glauber dynamics and low-delay csma scheduling," *IEEE Trans. on Information Theory*, vol. 58, no. 10, pp. 6541–6555, 2012.
[12] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *ACM SIGMETRICS/Performance*, Seattle, WA, June 2009.
[13] D. Shah and J. Shin, "Randomized scheduling algorithm for queueing networks," *Annals of Applied Probability*, vol. 22, no. 1, pp. 128–171, 2012.
[14] J. Ghaderi and R. Srikant, "On the design of efficient csma algorithms for wireless networks," arXiv report http://arxiv.org/abs/1003.1364, 2010.
[15] D. Shah and J. Shin, "Delay optimal queue-based CSMA," in *ACM SIGMETRICS*, Columbia University, NY, June 2010.
[16] M. Lotfinezhad and P. Marbach, "Throughput-optimal random access with order-optimal delay," in *Proceedings of IEEE INFOCOM*, April 2011.
[17] K.-K. Lam, C.-K. Chau, M. Chen, and S.-C. Liew., "Mixing time and temporal starvation of general csma networks with multiple frequency agility," in *IEEE ISIT*, July 2012.
[18] P.-K. Huang and X. Lin, "Improving the delay performance of CSMA algorithms: a virtual multi-channel approach," in *Proceedings of IEEE INFOCOM*, April 2013.
[19] C.-H. Lee, D. Y. Eun, S.-Y. Yun, and Y. Yi, "From glauber dynamics to metropolis algorithm: Smaller delay in optimal csma," in *IEEE ISIT*, July 2012.
[20] V. Subramanian and M. Alanyali, "Delay performance of CSMA in networks with bounded degree conflict graphs," in *IEEE ISIT*, July 2011.
[21] 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," *IEEE Trans. on Networking*, vol. 20, no. 3, pp. 825–836, June 2012.
[22] J. Ghaderi and R. Srikant, "The impact of access probabilities on the delay performance of q-csma algorithms in wireless networks," *IEEE/ACM Trans. on Networking*, vol. 21, no. 4, pp. 1063–1075, Aug. 2013.
[23] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. on Networking*, vol. 13, no. 2, pp. 411–424, Apr. 2005.
[24] S. P. Meyn and R. L. Tweedie, "Criteria for stability of Markovian processes I: Discrete time chains," *Advances in Applied Probability*, vol. 24, pp. 542–574, 1992.
[25] A. Ganesh, N. O'Connell, and D. Wischik, *Big Queues*. Springer, 2004.
[26] F. Bacelli and P. Brémaud, *Elements of Queueing Theory: Palm Martingale Calculus and Stochastic Recurrences*. Springer-Verlag, 2003.
[27] R. G. Addie and M. Zukerman, "An Approximation for Performance Evaluation of Stationary Single Server Queues," *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3150–3160, Dec. 1994.
[28] N. G. Duffield, "Exponential bounds for queues with Markovian arrivals," *Queueing Systems*, vol. 17, pp. 413–430, 1994.
[29] N. G. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single server queue, with application," *Proc. Cambridge Philos. Soc.*, vol. 118, pp. 363–374, 1995.
[30] F. Kelly, "Notes on effective bandwidths," *Stochastic networks: Theory and Applications*, Oxford University Press, 1996.
[31] T. P. Hayes and A. Sinclair, "A general lower bound for mixing of single-site dynamics on graphs," *Annals of Applied Probability*, vol. 17, no. 3, pp. 931–952, 2007.
[32] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*. monograph in preparation.
[33] C. J. Geyer, "Practical Markov Chain Monte Carlo," *Statistical Science*, vol. 7, no. 4, pp. 437–483, Nov. 1992.
[34] J. S. Liu, W. H. Wong, and A. Kong, "Covariance structure and convergence rate of the gibbs sampler with various scans," *Journal of the Royal Statistical Society, Ser. B.*, vol. 57, pp. 157–169, 1995.
[35] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Mathematical Society, 2009.
[36] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer-Verlag, 1999.
[37] R. M. Loynes, "The stability of a queue with non-independent inter-arrivals and service times," *Cambridge Phil.*, vol. 58, pp. 497–520, 1962.
[38] D. D. Botvich and N. Duffield, "Large deviations, the shape of the loss curve, and economies of scale in large multiplexers," *Queueing Systems*, vol. 20, pp. 293–320, 1995.
[39] A. Hordijk, Z. Liu, and D. Towsley, "Smoothing effect of the superposition of homogeneous sources in tandem networks," *Journal of Applied Probability*, vol. 37, no. 3, pp. 900–913, 2000.
[40] K. Kumaran, M. Mandjes, and A. Stolyar, "Convexity properties of loss and overflow functions," *Operations Research Letters*, vol. 31, no. 2, pp. 95 – 100, 2003.
[41] J. Choe and N. B. Shroff, "Use of supremum distribution of gaussian processes in queueing analysis with long-range dependence and self-similarity," in *Stochastic Models*, vol. 16, no. 2, Feb. 2000.
[42] D. Y. Eun and N. B. Shroff, "A measurement-analytic approach for QoS estimation in a network based on the dominant time scale," *IEEE/ACM Trans. on Networking*, vol. 11, no. 2, pp. 222–235, Apr. 2003.
[43] D. W. Stroock, *An Introduction to Markov Processes*. Springer, 2005.

PLACE PHOTO HERE

Jaewook Kwak received the B.E. degree in computer engineering from Hongik University, Seoul, Korea, in 2004 and the M.S. degree in computer engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006. He is currently pursuing the PhD degree in computer engineering at North Carolina State University, Raleigh, NC. His research interests include system modeling, performance analysis, and algorithm design in communications networks.

PLACE PHOTO HERE

Chul-Ho Lee received his B.E. degree (with honors) in Information and Telecommunication Engineering from Korea Aerospace University, Goyang, Korea, in 2003 and his M.S. degree in Information and Communications from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2005. He received his Ph.D. degree in Computer Engineering from North Carolina State University, Raleigh, NC, in 2012. He then worked as a postdoctoral research scholar in the Department of Electrical and Computer Engineering at North Carolina State University, Raleigh, NC, and a senior engineer in the the Digital Media & Communications (DMC) R&D Center at Samsung Electronics, Korea. He is currently an assistant professor in the Department of Electrical and Computer Engineering at Florida Institute of Technology, Melbourne, FL. His research interests include networking and performance evaluation, network data analytics, social network analysis, and mobile computing.

PLACE
PHOTO
HERE

Do Young Eun received his B.S. and M.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1995 and 1997, respectively, and Ph.D. degree from Purdue University, West Lafayette, IN, in 2003. Since August 2003, he has been with the Department of Electrical and Computer Engineering at North Carolina State University, Raleigh, NC, where he is currently an associate professor. His research interests include network modeling and performance analysis, mobile ad-hoc/sensor networks, mobility modeling, social networks, and graph sampling. He has been a member of Technical Program Committee of various conferences including IEEE INFOCOM, ICC, Globecom, ACM MobiHoc, and ACM Sigmetrics. He is currently on the editorial board of IEEE/ACM Transactions on Networking and Computer Communications Journal, and was TPC co-chair of WASA'11. He received the Best Paper Awards in the IEEE ICCCN 2005, IEEE IPCCC 2006, and IEEE NetSciCom 2015, and the National Science Foundation CAREER Award 2006. He supervised and co-authored a paper that received the Best Student Paper Award in ACM MobiCom 2007.