Provable Per-Link Delay-Optimal CSMA for General Wireless Network Topology

Dongmyung Lee†, Donggyu Yun†, Jinwoo Shin†, Yung Yi† and Se-Young Yun‡

***Abstract*—In the past few years, an exciting progress has been made on CSMA (Carrier Sense Multiple Access) algorithms that achieve throughput and utility optimality for wireless networks. However, most of these algorithms are known to exhibit poor delay performance making them impractical for implementation. Recently, several papers have addressed the delay issue of CSMA and yet, most of them are limited, in the sense that they focus merely on specific network scenarios with certain conditions rather than general network topology, achieve low delay at the cost of throughput reduction, or lack rigorous provable guarantees. In this paper, we focus on the recent idea of exploiting multiple channels (actually or virtually) for delay reduction in CSMA, and prove that it is *per-link delay order-optimal,* i.e., $O(1)$-asymptotic-delay per link, if the number of virtual channels is logarithmic with respect to mixing time of the underlying CSMA Markov chain. The logarithmic number is typically small, i.e., at most linear with respect to the network size. In other words, our contribution provides not only a provable framework for the multiple-channel based CSMA, but also the required explicit number of virtual-multi-channels, which is of great importance for actual implementation. The key step of our analytic framework lies in using quadratic Lyapunov functions in conjunction with (recursively applying) Lindley equation and Azuma's inequality for obtaining an exponential decaying property in certain queueing dynamics. We believe that our technique is of broad interest in analyzing the delay performances of other general queueing systems.**

I. INTRODUCTION

A. Motivation and Contribution

In wireless and computer networks, multiple nodes share a communication medium for transmitting their data packets. In order to achieve an efficient channel utilization to resolve any potential conflicts or interferences between competing nodes, designing a good scheduling algorithm, or medium access control (MAC) protocol is of crucial importance. In their seminal work, Tassiulas and Ephremides [1] proposed a scheduling algorithm known as 'Max-Weight' (MW) that achieves the throughput optimality. However, the MW algorithm requires to solve a NP-hard optimization problem in a centralized manner at each time slot, which is its main drawback for applying it to a large-scale network. Much efforts for designing simpler or distributed implementations of MW have been made since then, e.g., [2]–[4], but these algorithms, though distributed, require heavy message passing.

In the past few years or so, there has been a breakthrough, where the throughput/utility optimality can be achieved by just locally controlling the CSMA (Carrier Sense Multiple Access) parameters without explicit knowledge of neighboring information [5]–[9] hence providing a simple and distributed MAC with optimal performance guarantee. These algorithms are all based on one of the Markov Chain Monte Carlo (MCMC) methods called Glauber dynamics which can be used to sample the independent sets of a graph according to a product-form stationary distribution. However, it has been reported that the CSMA algorithms in general are known to suffer from poor delay performance [11]. Thus, it remains to develop or even verify the existence of such fully-distributed, yet highly delay-efficient MAC algorithms without message passing (possibly based on CSMA).

In this paper, we show the existence of a throughput-optimal, CSMA-based (thus fully-distributed) MAC with a provable *per-link* order-optimal delay performance for *general* wireless network topologies. To the best of our knowledge, this is the first such work in the literature, where even the network-wide order-optimal CSMA delay (which is weaker than our per-link optimality) is not known (see Section I-B for more details). In particular, this paper employs a virtual multi-channel CSMA, referred to as the *delayed* CSMA [17], and proves that it has the $O(1)$-asymptotic-delay *per link*, if the number of virtual channels is in the logarithmic order of the mixing time of the underlying CSMA Markov chain. The important implication of our result is that a small number of channels suffices to achieve the optimal delay performance of the CSMA algorithm since the logarithmic number is typically small, i.e., at most linear with respect to the network size [24]. The algorithm is easy to implement and fully-distributed, i.e., no message passing is necessary. The key part of our analytic framework is the use of quadratic Lyapunov functions in conjunction with (recursively applying) Lindley equation and Azuma's inequality for obtaining an exponential decaying property in certain queueing dynamics (see Section IV-A for more details). We believe that our technique is of broad interest in analyzing the delay performance of more general queueing systems.

B. Related Work

In the literature, several papers proposed different CSMA-based MACs, all of which are provably (close to be) delay-optimal under certain restrictions. Shah and Shin [10] proposed

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2013R1A2A2A01067633) and the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2012M3A6A6054195).

The first two authors are listed in alphabetical order of their last names. †: Department of Electrical Engineering, KAIST, Republic of Korea. ‡: School of EE, KTH, Sweden. (e-mails: {dmlee, dgyun}@lanada.kaist.ac.kr, jinwoos@kaist.ac.kr, yiyung@kaist.edu, seyoung@kth.se).

a modified queue-based CSMA algorithm where at each time instance a very small fraction of *frozen* nodes do not execute CSMA operations. By appropriately selecting such frozen nodes, the proposed algorithm leads to the network-wide $O(n)$ queue length, where n is the number of links. Lotfinezhad and Marbach [11] prove that by periodically resetting all links to become silent and then immediately restarting the classical CSMA protocol, the algorithm leads to the $O(1)$ per-packet delay for grid or torus interference graphs. Jiang et al. [12] study CSMA algorithms based on parallel Glauber dynamics. The authors prove that the algorithms can achieve the network-wide $O(n \log n)$ queue length, and Subramanian and Alanyali [13] further tightened the bound. The limitations in the above papers are *(i)* [10], [11] are only applicable to a specific type of topologies, i.e., geometric or torus (inference) network topologies, and *(ii)* [12], [13] require reduced offered loads and a tradeoff between throughput and delay occurs (i.e., in both papers, a certain amount of throughput should be sacrificed for low delay). We overcome these limitations in the current paper, and more importantly, we obtain the per-link $O(1)$ queue length, which is much stronger than the network-wide (or mean) queue length bound.

Another direction of designing CSMA algorithms for better delay performance is to use (actual or virtual) 'multi-channels,' which is in essence motivated by *resolving the temporal link starvation of CSMA via 'de-correlating' the temporal accessibility* to the wireless medium. Lam et al. [15] considers a CSMA algorithm with multiple frequency agility, such that more than one frequency channel is available yet a link can transmit on at most one of the channels. They associate temporal starvation to the mixing time and then show that the region of fast mixing time implies that of low temporal starvation through simulations. Huang and Lin [16] proposed an algorithm called VMC-CSMA in which multiple virtual channels (defined by dividing the time line) are used to emulate a multi-channel system and address the starvation problem. The algorithm randomly selects a virtual channel, and the schedule corresponding to this chosen channel is used at each time slot. The authors show that the expected packet delay for each link equals to the inverse of its long-term average rate, and the distribution of its head-of-line (HOL) waiting time can be asymptotically bounded. The multi-channel idea is also used in [17], where the authors propose the so-called *delayed* CSMA inducing multiple independent CSMA dynamics in a round robin manner. They showed that its asymptotic-delay performance can be improved by exploiting more channels (i.e., more rounds in the *delayed* CSMA). However, an explicit delay bound has yet to be studied, where especially a precise relation between the CSMA delay and the number of channels is practically important for implementation.

The question of designing MAC scheduling algorithms with low delay, not restricted to CSMA, in wireless networks has been also studied from a while ago. To name just a few, the MW algorithm empirically has a good delay performance, but its (network-wide or per-link) delay-optimality for general topology is not analytically known. Neely et al. [18] proved that maximal scheduling, which is suboptimal in terms of throughput, achieves $O(1)$ delay. Yi and Chiang [19] studied the 3-D tradeoff among delay, throughput, and complexity for a large class of queue-based scheduling schemes. A 'batching' policy, first considered by Neely et al. [20], is known to be almost delay-optimal for input-queued switch networks (a special topology of wireless networks), where its per-link delay is $O(\log n)$. Recently, Shah et al. [21] developed a centralized delay-optimal scheduling algorithm for wireless networks including input-queued switch networks. On the negative side, Shah et al. [22] showed that there exists no polynomial-time scheduler (including CSMA) with a polynomial delay for arbitrary network topologies unless $\mathbf{NP} \subseteq \mathbf{BPP}$. However, the authors consider the supremum of temporal delays over time, which does not imply that the asymptotic time-averaged delay performance is necessarily bad. Somewhat surprisingly, in the current paper we show that even a simple fully-distributed CSMA algorithm can achieve the asymptotic time-averaged $O(1)$ delay per link.

II. System Model

A. Model

We consider a network model where the interference relationship among the wireless links can be represented by the so-called *interference graph* $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ is the set of links, and $\mathcal{E}$ is the set of edges representing (symmetric) interference relationship between links. An edge $(i,j) \in \mathcal{E}$ exists between two links i and j if the corresponding wireless links interfere with each other. We denote $\mathcal{N}_v = \{w \in \mathcal{N} : (v,w) \in \mathcal{E}\}$ as the set of neighbors of node v. Time is divided into discrete slots, indexed by $t = 0, 1, \ldots$. Let $\boldsymbol{\sigma}(t) = (\sigma_v(t))_{v \in \mathcal{N}} \in \{0,1\}^{|\mathcal{N}|}$ be a *schedule* that represents the set of transmitting links at time t. A link v (or node v in the interference graph $\mathcal{G}$) is active if it is included in the schedule, i.e., $\sigma_v(t) = 1$, and is inactive otherwise. We denote by $\Omega \subseteq \{0,1\}^{|\mathcal{N}|}$ the set of all feasible schedules on $\mathcal{G}$, where a *feasible* schedule $\boldsymbol{\sigma}(t)$ should satisfy the independent set constraint i.e.,

$$\boldsymbol{\sigma}(t) \in \mathcal{I}(\mathcal{G}) \triangleq \{\boldsymbol{\chi} \in \{0,1\}^{|\mathcal{N}|} : \chi_v + \chi_w \leq 1, \forall (v,w) \in \mathcal{E}\}.$$

Each link is associated with a queue that has a dedicated exogenous arrival process. Let $A_v(t)$ denote the number of packet (of unit-size) arrival of link v at time t. We assume Bernoulli arrivals, i.e., $A_v(t) \in \{0,1\}$ with $\mathbb{P}[A_v(t) = 1] = \lambda_v$. Let $\boldsymbol{Q}(t) = (Q_v(t))_{v \in \mathcal{N}}$ denote the vector of queue sizes at time t. Then it has the following dynamics:

$$Q_v(t) = \max\{Q_v(t-1) + A_v(t) - \sigma_v(t), 0\}, \quad t \geq 1.$$

where it is called the *Lindley* equation. We denote by $A_v[t,t']$ and $D_v[t,t']$ the number of arrivals and (potential) departures in the interval $[t,t']$, respectively, i.e.,

$$A_v[t,t'] = \sum_{s=t}^{t'} A_v(s) \qquad \text{and} \qquad D_v[t,t'] = \sum_{s=t}^{t'} \sigma_v(s).$$

In addition, we let $\Delta_v[t,t'] = A_v[t,t'] - D_v[t,t']$. Then, by recursively applying the Lindley equations, we obtain the following:

$$\begin{aligned}Q_v(t) &= \max\{Q_v(t-1) + A_v[t,t] - D_v[t,t], 0\} \\ &= \max\{\max\{Q_v(t-2) + A_v[t-1,t-1] - \\ &\quad D_v[t-1,t-1], 0\} + A_v[t,t] - D_v[t,t], 0\} \\ &\quad \cdots \\ &= \max\Big\{Q_v(0) + A_v[1,t] - D_v[1,t], \max_{2\le s\le t} \\ &\qquad \{A_v[s,t] - D_v[s,t]\}, 0\Big\}. \quad (1)\end{aligned}$$

The above equation will play a crucial role in our analysis.

We define the capacity region $\mathcal{C} \subset [0,1]^{|\mathcal{N}|}$ of a network as the convex hull of the feasible scheduling set $\mathcal{I}(\mathcal{G})$, i.e.,

$$\mathcal{C} = \left\{ \sum_{\chi\in\mathcal{I}(\mathcal{G})} \alpha_\chi \chi : \sum_{\chi\in\mathcal{I}(\mathcal{G})} \alpha_\chi = 1, \alpha_\chi \ge 0, \forall \chi \in \mathcal{I}(\mathcal{G}) \right\}.$$

Let $\boldsymbol{\lambda} = [\lambda_v]$ and it is called *admissible* if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$, where

$$\boldsymbol{\Lambda} = \left\{ \boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{N}|} : \lambda_v \le \gamma_v,\ \forall v,\ \text{for some } \boldsymbol{\gamma} = [\gamma_v] \in \mathcal{C} \right\}.$$

The intuition behind this notion of $\boldsymbol{\Lambda}$ comes from the fact that any scheduling algorithm has to choose a schedule from $\mathcal{I}(\mathcal{G})$ at each time and hence the arrival rate must belong to $\boldsymbol{\Lambda}$ (otherwise, queues should grow over time). In addition, for given $\varepsilon > 0$, $\boldsymbol{\lambda}$ is called ε-*admissible* if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^\varepsilon$, where

$$\boldsymbol{\Lambda}^\varepsilon = \left\{ \boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{N}|} : \lambda_v + \varepsilon < \gamma_v,\ \forall v,\ \text{for some } \boldsymbol{\gamma} = [\gamma_v] \in \mathcal{C} \right\}.$$

Finally, $\boldsymbol{\lambda}$ is called *strictly admissible* if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^\circ$ where $\boldsymbol{\Lambda}^\circ = \bigcup_{\varepsilon>0} \boldsymbol{\Lambda}^\varepsilon$.

B. Performance Metric

A scheduling algorithm decides a sequence of $\boldsymbol{\sigma}(t) \in \mathcal{I}(\mathcal{G})$ over $t = 0, 1, \ldots$. In this section, we introduce the main performance metrics, which are the throughput and delay of scheduling algorithms. First, we define the throughput-optimality:

Definition 2.1 (Throughput-Optimality): A scheduling algorithm is called *throughput-optimal* if for any strictly admissible arrival rate $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^\circ$, queues remain finite with probability 1 under the algorithm, i.e.,

$$\limsup_{t\to\infty} \sum_v Q_v(t) < \infty, \qquad \text{with probability 1.} \quad (2)$$

A popular approach for showing the throughput-optimality is (a) defining the underlying network Markov chain induced by a scheduling algorithm and (b) proving its *positive recurrence*. We now introduce the delay optimality studied in this paper.

Definition 2.2 (Delay-Optimality): A scheduling algorithm is called *per-link delay-optimal* (or simply delay-optimal) [1], if for any ε-admissible arrival rate $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^{\boldsymbol{\varepsilon}}$ with $\varepsilon = \omega(1)$,

$$\limsup_{t\to\infty} \mathbb{E}\left[Q_v(t)\right] = O(1), \qquad \text{for all } v \in \mathcal{N}.$$

In the above definition, the orders $\omega(1)$ and $O(1)$ are with respect to the network size $|\mathcal{N}|$, i.e., delay-optimality means that the *per-link* queue-size remains 'constant' as the network size grows.

[1]This per-link optimality is much stronger than the 'network-wide' optimality defined by the averaged delay over all links.

C. Delayed CSMA

Our interest is in fully-distributed CSMA scheduling algorithms, where in particular, we study the *delayed* CSMA proposed in [17]. The main idea is to use multiple schedulers in a round-robin manner in order to effectively reduce the correlations between the link state process, in an attempt to alleviate the so-called *starvation problem*, i.e., once a schedule is chosen, it keeps being scheduled without any change for a large number of slots. This algorithm is formally stated in Algorithm 1.

Algorithm 1 Delayed CSMA [17]

```
Initialize: for all links v ∈ N, σ_v(t) = 0, t = 0,...,T − 1.
At each time t ≥ T: links find a decision schedule,
D(t) ∈ I(G) through a randomized procedure, and
for all links v ∈ D(t) do
  if Σ_{w∈N_v} σ_w(t − T) = 0 then
    σ_v(t) = 1 with probability r_v/(1+r_v)
    σ_v(t) = 0 with probability 1/(1+r_v)
  else
    σ_v(t) = 0
  end if
end for
for all links w ∉ D(t) do
    σ_v(t) = σ_v(t − T)
end for
```

In the *delayed* CSMA, at each time slot, a decision schedule is chosen $\mathcal{D}(t) \in \mathcal{I}(\mathcal{G})$, which corresponds to a selection of an independent set of $I(\mathcal{G})$. The active links in the decision schedule become the candidate links which may change their state. There are various ways to choose a decision schedule $\mathcal{D}(t) \in \mathcal{I}(\mathcal{G})$ at each time slot. For example, each link simply attempts to access the medium with a fixed access probability a_v and then $v \in \mathcal{D}(t)$ with probability $a_v \prod_{u\in\mathcal{N}_v}(1-a_u)$, or a randomized scheme with light control message exchanges can be used, as in [25]. In general, we assume that $\{\mathcal{D}(t)\}$ is a set of independent identical random variables such that $\mathbb{P}[v \in \mathcal{D}(t)] > 0$ for all v. Under the assumption, given the 'fugacity' $[r_v]$, the schedule $\{\boldsymbol{\sigma}(t) : t \equiv k \pmod{T}\}$ forms a (discrete-time) irreducible and aperiodic Markov chain for $k = 0, 1, \ldots, T-1$, e.g., the k-th Markov chain is $\{\boldsymbol{\sigma}(uT + k) : u = 0, 1, 2, \ldots\}$. The common stationary distribution $\boldsymbol{\pi} = [\pi_{\boldsymbol{\sigma}}]$ is given by

$$\pi_{\boldsymbol{\sigma}} = \frac{1}{Z} \prod_{v\in\mathcal{N}} r_v^{\sigma_v}, \quad (3)$$

where $Z = \sum_{\boldsymbol{\sigma}\in\Omega} \prod_{v\in\mathcal{N}} r_v^{\sigma_v}$ is a normalizing constant. Hence, one can think that the algorithm utilizes multiple T independent Markov chains (or schedulers). From their ergodicities, we know that for all $v \in \mathcal{N}$,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t-1} \sigma_v(s) = \mathbb{P}_\pi[\sigma_v = 1].$$

The essence of recently developed throughput-optimal CSMA algorithms [7], [9] is finding an appropriate fugacity $[r_v]$ such that the long-term link throughput $\lim_{t\to\infty} \frac{1}{t}\sum_{s=0}^{t-1} \sigma_v(s)$ is (strictly) greater than the arrival rate λ_v (whenever $\boldsymbol{\lambda}$ is strictly admissible). Given such a fugacity, it is not hard to prove the desired positive recurrence of the underlying network Markov chain [7]. Several authors proposed different adaptive updating rules on fugacities, which converge to the appropriate fugacity [7]–[9], [23]. Since the main focus of this paper is to analyze the delay performance of the *delayed* CSMA, we assume that links initially start with the desired fugacity. Formally speaking, for given ε-admissible arrival rate $\boldsymbol{\lambda}$, we assume that

$$\lim_{t\to\infty}\frac{1}{t}\sum_{s=0}^{t-1}\sigma_v(s)=\mathbb{P}_\pi[\sigma_v=1]\geq\lambda_v+\varepsilon,\ \text{for all } v\in\mathcal{N}. \quad (4)$$

This is merely for simple presentation of our proof, and one can easily extend it without the assumption, i.e., under the known adaptive fugacity updating rules in [7]. This is because our delay metric is defined in an asymptotic manner (i.e., uses 'limsup') and the exact convergence in the fugacity updating rules is not crucial in our proof strategy.[2]

III. Main Result

This section presents the main result of this paper, namely the delay-optimality of the *delayed* CSMA algorithm. We first summarize the main contribution of our paper beyond [17]. The authors have shown that under *delayed* CSMA, the probability that a queue length is larger than some value x, is approximated by a function that is exponentially decreasing in x and T. However, the result is quite asymptotic and does not imply delay-optimality, i.e., it is not clear whether the expected queue length becomes $O(1)$ as T grows. Furthermore, a more precise relation between the queue length and T is of significant importance for its actual implementation, which we characterize in this paper using the mixing time of the underlying CSMA Markov chain $\{\boldsymbol{\sigma}(t):t\equiv k \pmod T\}$.

A. Preliminaries: Mixing Time

To describe our result formally, we first introduce the necessary definitions of the *total variation distance* and the corresponding *mixing time* of the CSMA Markov chain. The total variation distance between two probability distributions $\boldsymbol{\eta}=[\eta_i]$ and $\boldsymbol{\nu}=[\nu_i]$ on state space Ω is

$$\|\boldsymbol{\eta}-\boldsymbol{\nu}\|_{TV}=\frac{1}{2}\sum_{i\in\Omega}|\eta_i-\nu_i|.$$

Using this distance metric, the *mixing time* of the k-th CSMA Markov chain $\{\boldsymbol{\sigma}(uT+k):u=0,1,2,\dots\}$ is defined as follows:

$$M^{(k)}(\delta)=\inf\Big\{s:\max_{\boldsymbol{\mu}(k)}\big\|\boldsymbol{\mu}(uT{+}k)-\boldsymbol{\pi}\big\|_{TV}\leq\delta,\ \forall u\geq s\Big\},$$

where $\delta>0$ is some constant (which we will choose later) and $\boldsymbol{\mu}(t)$ denotes the probability distribution of random variable $\boldsymbol{\sigma}(t)$. The mixing time measures how long it takes for the k-th CSMA Markov chain to converge to the stationary distribution for arbitrary initial distribution $\boldsymbol{\mu}(k)$. Since we assume the fixed common fugacity across the Markov chains, the mixing time $M^{(k)}(\delta)$ is identical for $k=0,1,\dots,T-1$. Hence, we use $M(\delta)=M^{(k)}(\delta)$.

[2]The 'fixed' fugacity assumptions have also been taken in other papers dealing with CSMA delay, e.g., [12].

B. Main Result: Delay-Optimality

Now we are ready to state the main result of this paper, i.e., the delay-optimality of the *delayed* CSMA algorithm.

Theorem 3.1: For any ε-admissible arrival rate $\boldsymbol{\lambda}\in\boldsymbol{\Lambda}^{\varepsilon}$, there exists $T^*=O\left(\frac{1}{\varepsilon^3}\log M(\varepsilon/4)\right)$ such that for all $T>T^*$, the corresponding *delayed* CSMA algorithm is delay-optimal, more formally,

$$\lim_{t\to\infty}\mathbb{E}[Q_v(t)]=O\left(\frac{1}{\varepsilon^5}\right),\qquad\text{for all } v\in\mathcal{N}.$$

The above theorem states that the per-link average queue-size is bounded by a constant for sufficiently large T, the number of independent CSMA schedulers. The purpose of choosing large T is to effectively reduce the dependency among consecutive link states, which promotes much faster link state changes and hence alleviates the starvation problem.

We further remark that Theorem 3.1 is optimal with respect to the network size, but not with respect to ε, e.g., the best order of delay in both parameters is $O(1/\varepsilon)$. In this paper, we do not make much efforts to optimize our analysis for the better delay dependency in ε and the tighter bound of T^*. For example, in our simulation results (see Section V), we observe that $T=2$ is enough for the order-optimal delay for grid-like graphs. We provide the proof of Theorem 3.1 in the following section.

IV. Proofs: Theorem 3.1 and Necessary Lemmas

A. Proof Strategy

We first describe our proof strategy at a high level, followed by the detailed proof in Section IV-B. We use the popular approach using a quadratic Lyapunov function to prove the desired delay bound in Theorem 3.1. In this approach, one has to define an appropriate network Markov chain and show a certain negative drift property for the Lyapunov function (see Lemma 4.1) . In particular, the network Markov chain $\{X(t)\}$ under the *delayed* CSMA algorithm is

$$X(t)=\big(\boldsymbol{Q}(t),\boldsymbol{\sigma}(t),\boldsymbol{\sigma}(t-1),\dots,\boldsymbol{\sigma}(t-T+1)\big).$$

Two major technical challenges for deriving the desired negative drift property are:

- It is necessary that the CSMA Markov chains mix, which takes the mixing time $M=M(\varepsilon/4)$. Otherwise, the scheduling dynamics is hard to analyze.
- Even after the CSMA Markov chains mix, schedules $\{\boldsymbol{\sigma}(\cdot)\}$ are correlated (i.e., they are not i.i.d random variables), but one has to show that they still satisfy some version of law of large numbers.

For the first issue, we design a stopping time obtaining the negative drift property to be much later than the mixing time so that the negative drift in the mixing period dominates the (potential) positive drift in the initial non-mixing one. For the second issue, we first observe that schedules $\{\boldsymbol{\sigma}(\cdot)\}$ in each time interval of length T are almost i.i.d random variables (after the CSMA Markov chains mix) due to the design of the delayed CSMA algorithm. Hence, we choose T large enough so that (a) the 'variance' of the sum of schedules in each time interval of length T is small enough (b) the possible correlation across the time intervals is compensated. Specifically, for (a) we use Azuma's inequality for obtaining an exponential decaying property of queueing dynamics in each time interval of length T, and for (b) the union bound is used under Lindley recursions.

B. Proof of Theorem 3.1

We now start toward proving Theorem 3.1. First observe that the network Markov chain $\{X(t)\}$ is aperiodic and irreducible. Its ergodicity (i.e., positive recurrence) can be derived using the following key lemma whose proof is presented in Section IV-C.

***Lemma** 4.1:* For any ε-admissible arrival rate $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^{\varepsilon}$, there exist positive numbers $T^* = O\left(\frac{1}{\varepsilon^3}\log M\right)$ such that for all $T > T^*$ and $v \in \mathcal{N}$,

$$\mathbb{E}\big[Q_v(t+7MT/\varepsilon)^2 \,|\, X(t)\big] \le Q_v(t)^2 - \frac{Q_v(t)}{2} + O\left(\frac{1}{\varepsilon^5}\right).$$

The positive recurrence of $\{X(t)\}$ follows easily via the Lyapunov-Foster criteria with Lemma 4.1 and quadratic Lyapunov function $L(X(t)) = \sum_v Q_v(t)^2$. Furthermore, taking expectations with respect to the distribution of $X(t)$ in both sides of the conclusion of Lemma 4.1, we have

$$\mathbb{E}[Q_v(t+7MT/\varepsilon)^2] \le \mathbb{E}[Q_v(t)^2] - \frac{\mathbb{E}[Q_v(t)]}{2} + O\left(\frac{1}{\varepsilon^5}\right).$$

Due to the fact that $\lim_{t\to\infty}\mathbb{E}[Q_v(t+7MT/\varepsilon)^2] = \lim_{t\to\infty}\mathbb{E}[Q_v(t)^2]$ from the ergodicity of $\{X(t)\}$, we obtain the conclusion of Theorem 3.1: $\lim_{t\to\infty}\mathbb{E}[Q_v(t)] = O\left(\frac{1}{\varepsilon^5}\right)$.

C. Proof of Lemma 4.1

We first present the following lemma which plays a crucial role for proving Lemma 4.1.

***Lemma** 4.2:* For any $Q \ge 0$ satisfying $\varphi \ge Q - MT/2$, it follows that

$$\mathbb{P}\Big[Q_v\big(t+7MT/\varepsilon\big) \ge \varphi \,\big|\, \mathcal{L}\Big] \le g(T,\varphi),$$

where $g(T,\varphi) = \frac{140M}{\varepsilon^3}\exp\left(-\frac{\varepsilon^2 T}{80}\right) + \frac{18}{\varepsilon^2}\exp\left(-\frac{\varphi\varepsilon}{9}\right)$ and

$$\mathcal{L} = \{Q_v(t) = Q, \boldsymbol{\sigma}(t), \dots, \boldsymbol{\sigma}(t-T+1)\}.$$

The proof of the above lemma is provided in Section IV-D. Lemma 4.2 implies that given a network state at time t, the distribution of the (per-link) queue length at time $t+7MT/\varepsilon$ has an exponential decaying property (i.e., light tail). We will use this property to bound the expected quadratic queue length at time $t+7MT/\varepsilon$ for completing the proof of Lemma 4.1.

We proceed the proof by studying two disjoint cases: ***Case (i):*** $Q_v(t) = Q \le C$ and ***Case (ii):*** $Q_v(t) = Q > C$, where $\varepsilon^2 C = O(1)$ is some large constant.

Case (i): $Q_v(t) = Q \le C$.

In this case, we obtain

$$\begin{aligned}
&\mathbb{E}\left[Q_v(t+7MT/\varepsilon)^2 - Q_v(t)^2 \,|\, \mathcal{L}\right] \\
&\quad\le \mathbb{E}\left[Q_v(t+7MT/\varepsilon)^2 \,|\, \mathcal{L}\right] - Q_v(t) \\
&\quad\overset{(a)}{\le} \sum_{i=1}^{(C+7MT/\varepsilon)^2} \mathbb{P}\Big[Q_v(t+7MT/\varepsilon)^2 \ge i \,\Big|\, \mathcal{L}\Big] - Q_v(t) \\
&\quad= \sum_{i=1}^{(C+7MT/\varepsilon)^2} \mathbb{P}\Big[Q_v(t+7MT/\varepsilon) \ge \sqrt{i} \,\Big|\, \mathcal{L}\Big] - Q_v(t) \\
&\quad\overset{(b)}{\le} \sum_{i=1}^{(C+7MT/\varepsilon)^2} g\left(T, \sqrt{i}\right) - Q_v(t),
\end{aligned}$$

where (a) is because

$$Q_v(t+7MT/\varepsilon) \le Q_v(t) + 7MT/\varepsilon \le C + 7MT/\varepsilon,$$

and we apply Lemma 4.2 for (b) since $Q - MT/2 < 0$, i.e., $Q \le C = O(1/\varepsilon^2)$ and $T > T^* = \Omega\left(\frac{1}{\varepsilon^2}\right)$. Hence, we have

$$\begin{aligned}
&\mathbb{E}\left[Q_v(t+7MT/\varepsilon)^2 - Q_v(t)^2 \,|\, \mathcal{L}\right] \\
&\le \sum_{i=1}^{(C+7MT/\varepsilon)^2} g\left(T, \sqrt{i}\right) - Q_v(t) \\
&= \sum_{i=1}^{(C+7MT/\varepsilon)^2} \left[\frac{O(M)}{\varepsilon^3}\exp\left(-\frac{\varepsilon^2 T}{80}\right) \right. \\
&\qquad\qquad \left. + \frac{O(1)}{\varepsilon^2}\exp\left(-\frac{\varepsilon\sqrt{i}}{9}\right)\right] - Q_v(t) \\
&\le O\left(\frac{M^3T^2}{\varepsilon^5} + \frac{CM^2T}{\varepsilon^4} + \frac{C^2M}{\varepsilon^3}\right)\exp\left(-\frac{\varepsilon^2 T}{80}\right) \\
&\qquad\qquad + \sum_{i=1}^{\infty}\frac{O(1)}{\varepsilon^2}\exp\left(-\frac{\varepsilon\sqrt{i}}{9}\right) - Q_v(t) \\
&\overset{(c)}{=} O(1) + \frac{O(1)}{\varepsilon^2}\underbrace{\sum_{i=1}^{\infty}\exp\left(-\frac{\varepsilon\sqrt{i}}{9}\right)}_{(*)} - Q_v(t),
\end{aligned} \tag{5}$$

where (c) is due to large enough T with $T > T^* = \Omega\left(\frac{1}{\varepsilon^3}\log M\right)$ and $C = O(1/\varepsilon^2)$. Therefore, it suffices to bound the term $(*)$ for the proof of Lemma 4.1. To this end, consider large enough i^* with $i^* = O(1/\varepsilon^3)$ so that $\sqrt{i} > \frac{18}{\varepsilon}\log i$ for all $i \ge i^*$. Then, it follows that

$$\begin{aligned}
&\sum_{i=1}^{\infty}\exp\left(-\frac{\varepsilon\sqrt{i}}{9}\right) = \sum_{i=1}^{i^*}\exp\left(-\frac{\varepsilon\sqrt{i}}{9}\right) + \\
&\qquad \sum_{i=i^*+1}^{\infty}\exp\left(-\frac{\varepsilon\sqrt{i}}{9}\right) \le i^* + \sum_{q=1}^{\infty}\frac{1}{i^2} = O\left(\frac{1}{\varepsilon^3}\right).
\end{aligned} \tag{6}$$

Combining (5) and (6) completes the proof of Lemma 4.1 for ***Case (i)***.

Case (ii): $Q(t) = Q > C$. In this case, we define the following (sub-)event of $\mathcal{L}$:

$$\mathcal{L}_- = \{Q_v(t+7MT/\varepsilon) - Q_v(t) \le -1\} \cap \mathcal{L}.$$

Using this notation, we have

$$\begin{aligned}
&\mathbb{E}\left[Q_v(t+7MT/\varepsilon)^2 - Q_v(t)^2 \,|\, \mathcal{L}\right] \\
&= \mathbb{E}\left[(Q(t+7MT/\varepsilon) - Q(t))(Q(t+7MT/\varepsilon) + Q(t)) \,|\, \mathcal{L}\right] \\
&= \mathbb{P}[\mathcal{L}_- \,|\, \mathcal{L}] \cdot \mathbb{E}\left[-Q(t+7MT/\varepsilon) - Q(t) \,|\, \mathcal{L}_-\right] \\
&\quad + \sum_{i=0}^{7MT/\varepsilon} \mathbb{P}[Q(t+7MT/\varepsilon) - Q(t) = i \,|\, \mathcal{L}] \cdot \left(i^2 + 2iQ(t)\right) \\
&\le -Q(t) \cdot \mathbb{P}[\mathcal{L}_- \,|\, \mathcal{L}]
\end{aligned}$$

$$
\begin{aligned}
&+ 3Q(t) \sum_{i=0}^{7MT/\varepsilon} i^2 \mathbb{P}[Q(t+7MT/\varepsilon) - Q(t) = i \,|\, \mathcal{L}] \\
&\le -Q(t) \cdot \Big(1 - g(T, Q(t)-1)\Big) \\
&\quad + 3Q(t) \sum_{i=1}^{49M^2T^2/\varepsilon^2} \mathbb{P}[Q(t+7MT/\varepsilon) - Q(t) \ge \sqrt{i} \,|\, \mathcal{L}] \\
&\le \Big(\underbrace{g(T, Q(t)-1) + 3 \sum_{i=1}^{49M^2T^2/\varepsilon^2} g\Big(T, Q(t)+\sqrt{i}\Big)}_{(**)} - 1\Big) Q(t),
\end{aligned}
\tag{7}
$$

where we apply Lemma 4.2 and one can show that the term $(**)$ is smaller than 1/2 if $Q(t) > C$ and $\varepsilon^2 C = O(1)$ is a large enough constant, similarly as we bound the term $(*)$ in ***Case (i)***:

$$
g(T, Q(t)-1) + 3 \sum_{i=1}^{49M^2T^2/\varepsilon^2} g\Big(T, Q(t)+\sqrt{i}\Big) \;\le\; \frac{1}{2}. \tag{8}
$$

The complete proof of the above inequality is given in the Appendix. This completes the proof of Lemma 4.1 for ***Case (ii)***.

D. Proof of Lemma 4.2

From the Lindley recursion (1) and the union bound, we have

$$
\begin{aligned}
&\mathbb{P}[Q_v(t+7MT/\varepsilon) \ge \varphi \,|\, \mathcal{L}] \\
&\le \underbrace{\mathbb{P}[Q(t+MT) + \Delta_v[t+MT+1, t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}]}_{(\dagger)} \\
&\quad + \underbrace{\mathbb{P}\Big[\max_{MT+2 \le s \le 7MT/\varepsilon} \Delta_v[t+s, t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}\Big]}_{(\ddagger)}.
\end{aligned}
$$

Hence, it suffices to bound both terms $(\dagger)$ and $(\ddagger)$ for the proof of Lemma 4.2, which we present in what follows.

Bound for $(\dagger)$**.** We will use Azuma's inequality and the union bound for obtaining the bound of $(\dagger)$. To this end, we define the appropriate martingale: for $s \in (t+(i-1)T, t+iT]$ and $i = M+1, M+2, \dots$,

$$
Y_s^v = \sum_{u=t+(i-1)T+1}^{s} (\Delta_v[u,u] + \varepsilon/2),
$$

and initially $Y_{t+(i-1)T}^v = 0$. Then, using the definition of the mixing time $M = M(\varepsilon/4)$, the conditional independence of $\{\Delta_v[s,s] : s \in [t+(i-1)T, t+iT]\}$, and the fact that $\mathbb{E}[\Delta_v[s,s] + \varepsilon/2] \le 2\cdot(\varepsilon/4) - \varepsilon + \varepsilon/2 = 0$, it is not hard to check that the following random variables form a supermartingale:

$$
\{Y_s^v : s \in [t+(i-1)T, t+iT]\}.
$$

Using this notation, we have

$$
\begin{aligned}
&\mathbb{P}[Q(t+MT) + \Delta_v[t+MT+1, t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}] \\
&\le \mathbb{P}[Q(t) + MT + \Delta_v[t+MT+1, t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}] \\
&= \mathbb{P}[\Delta_v[t+MT+1, t+7MT/\varepsilon] \ge \varphi - Q(t) - MT \,|\, \mathcal{L}] \\
&= \mathbb{P}\Big[\sum_{i=M+1}^{7M/\varepsilon} \big(Y_{t+iT}^v - Y_{t+(i-1)T}^v - \frac{\varepsilon T}{2}\big) \ge \varphi - Q - MT \,\Big|\, \mathcal{L}\Big] \\
&\le \mathbb{P}\Big[\bigcup_{i=M+1}^{7M/\varepsilon} \{Y_{t+iT}^v - Y_{t+(i-1)T}^v - \frac{\varepsilon T}{2} \ge \frac{\varphi - Q - MT}{M(7/\varepsilon - 1)}\} \,|\, \mathcal{L}\Big] \\
&\overset{(a)}{\le} \sum_{i=M+1}^{7M/\varepsilon} \mathbb{P}[Y_{t+iT}^v - Y_{t+(i-1)T}^v \ge \frac{\varepsilon T}{2} - \frac{\varepsilon T}{4} \,\Big|\, \mathcal{L}] \\
&\overset{(b)}{\le} \sum_{i=M+1}^{7M/\varepsilon} \exp\left(\frac{-(\frac{\varepsilon T}{4})^2}{2\sum_{r=1}^{T}(1+\varepsilon/2)^2}\right) \\
&\le \frac{7M}{\varepsilon} \exp\left(\frac{-(\frac{\varepsilon T}{4})^2}{5T}\right) = \frac{7M}{\varepsilon} \exp\left(-\frac{\varepsilon^2 T}{80}\right),
\end{aligned}
\tag{9}
$$

where (a) is due the the union bound and

$$
\frac{\varphi - Q(t) - MT}{M(7/\varepsilon - 1)} \ge \frac{-3MT/2}{M(7/\varepsilon - 1)} \ge \frac{-3MT/2}{6M/\varepsilon} = -\frac{\varepsilon T}{4},
$$

and (b) is due to Azuma's inequality on the submartingale $\{Y_s^v : s \in [t+(i-1)T, t+iT]\}$. Note that the bound (9) is identical for all $i = M+1, M+2, \dots$.

Bound for $(\ddagger)$**.** One can bound $(\ddagger)$ similarly as we did for $(\dagger)$. In particular, we obtain

$$
\begin{aligned}
&\mathbb{P}\Big[\max_{MT+2 \le s \le 7MT/\varepsilon} \Delta_v[t+s, t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}\Big] \\
&\qquad \le \frac{133M}{\varepsilon^3} \exp\left(-\frac{\varepsilon^2 T}{80}\right) + \frac{18}{\varepsilon^2} \exp\left(-\frac{\varphi\varepsilon}{9}\right).
\end{aligned}
\tag{10}
$$

We provide the complete proof of (10) in the Appendix. Combining (9) and (10) leads to the desired conclusion of Lemma 4.2.

V. Simulation Results

A. Setup

Interference graphs. In this section, we provide simulation results to verify our analytical findings. For the simulated network topology, as depicted in Fig. 1, we use grid-like graphs, which have been popularly used as representative interference graphs. We consider $n \times n$ torus interference graphs composed of total number of n^2 links ($n \ge 2$), where every link has exactly four interfering neighbors.

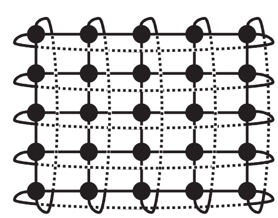
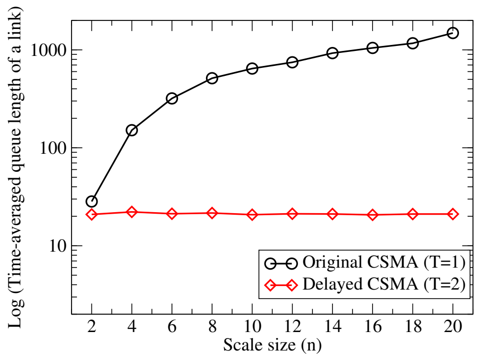
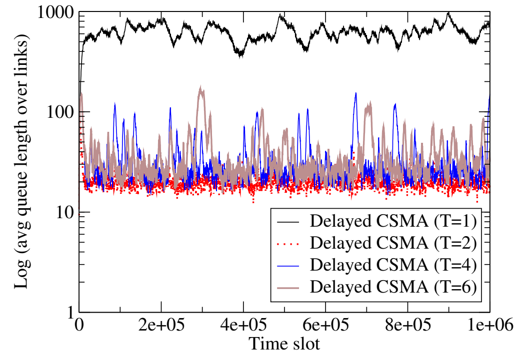


Fig. 1. 5×5 torus interference graph

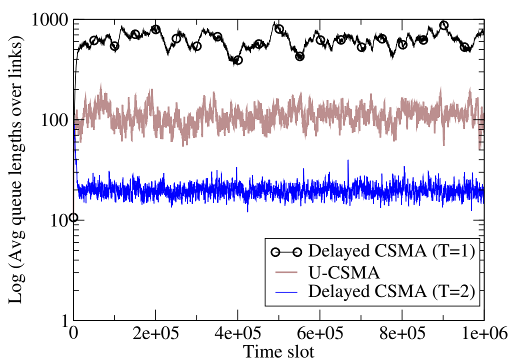
Loads and fugacity. We vary the scale size n of the torus graph to see how the delay performance (measured by the queue lengths) behaves. For a given arrival rate, we apply the appropriate fugacity that leads to the service rates over



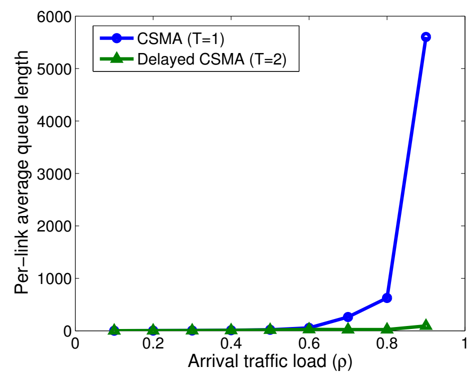
(a) Average queue length (over both time slots and links) vs. scale size n in $n \times n$ torus graphs: y-axis is log-scaled.



(b) Average queue length traces (over links) for the *delayed* CSMAs with different values of T ($n = 10$).



(c) Average queue length traces (over links) for the *delayed* CSMA with $T = 1, 2$ and U-CSMA ($n = 10$).



(d) Average queue length (over both time slots and links) for different values of loads ($n = 10$).

Fig. 2. Simulation results for the torus interference graphs.

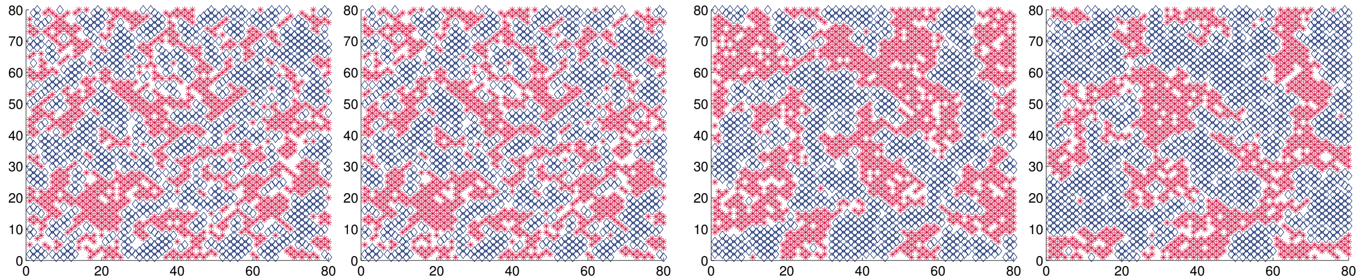
link that guarantees stability, by running one of the adaptive algorithms, e.g., [7] that searches for the target fugacity. In all plots except for Fig. 2(d) (which shows the delay performance for varying loads), we use a reasonably high load $\rho = 80\%$ i.e. $\lambda_v = 0.5 \times 0.8 = 0.4$ for every link v, where note that $\lambda_v = 0.5$ for all v corresponds to the boundary of the capacity region. We run 10^6 slots for all simulations. In the *delayed CSMA*, we use access probability $a_v = 0.25$ in choosing decision schedules, but we remark that other values of a_v show similar trends to what is presented here.

Compared algorithm: U-CSMA [11]. We also compare the *delayed* CSMA for various T with U-CSMA [11] that is provably delay-optimal in torus graphs. We comment that the original U-CSMA has been developed under a continuous time framework, which means that a procedure of determining decision schedules is not required. Recall that the key idea of U-CSMA lies in resetting the underlying CSMA Markov chain with a given period, say P. A naive candidate discrete-time version of U-CSMA is just the *delayed* CSMA with $T = 1$, that restarts itself every P time slots. However, our simulation experience of such a version of U-CSMA showed that it performs significantly poorly, because the step of finding decision schedules makes U-CSMA start with a very small number of active links at each reset period, which leads to scheduling only a small number of links during each period. This problem can be relaxed by enlarging P, but then it weakens the effect of resetting. Thus, for a fair comparison, we employ an almost "ideal" version of U-CSMA which magically has a good decision schedule, i.e., its number of links in a decision schedule is close to that of a maximum independent set.

B. Results

We first verify the delay optimality proved in Theorem 3.1. Fig. 2(a) shows the average queue-size vs. the scale size n in the torus. Indeed, we observe that while queue size of the "classical CSMA" (i.e., $T = 1$ in the *delayed* CSMA) linearly increases with n, that of the *delayed* CSMA for $T = 2$ does not increase with n, i.e., $O(1)$ delay. We comment that at least in the tested torus topology, just $T = 2$ suffices to achieve very low delays, even for highly large scales, e.g., $n = 20$ (thus 400 links). This is highly valuable in practice, because a small T significantly simplifies practical implementation. Fig. 2(b) shows the average queue size traces over time for various values of $T = 1, 2, 4, 6$, where we observe significant difference between $T = 1$ and other values of T's, whereas marginal difference among $T > 2$ is observed.

We now compare the *delayed* CSMA with U-CSMA in Fig. 2(c), where we have plotted the queue size traces for $T = 1, 2$. For U-CSMA, as indicated in [11], we use



(a) Time $t = 1999$ for $T = 1$ (b) Time $t = 2000$ for $T = 1$ (c) Time $t = 1999$ for $T = 2$ (d) Time $t = 2000$ for $T = 2$

Fig. 3. Snapshots of the schedules of the *delayed* CSMA with $T = 1$ ((a) and (b)) and $T = 2$ ((c) and (d)) for 80×80 torus. (A magenta star represents an even active link and an indigo diamond represents an odd active link.)

$P = 100$ for the reset period. First, we observe that U-CSMA outperforms the *delayed* CSMA with $T = 1$, which is because resetting reduces the correlations among schedules over time. However, we also see that the *delayed* CSMA with $T = 2$ is significantly better than U-CSMA (recall that y-axis is log-scaled). This demonstrates that an approach of weakening temporal correlations by running multiple Markov chains leads to much higher delay performance gain than that of re-starting new Markov chains periodically. The simulation result for different values of loads in Fig. 2(d) also shows a significant decrease of delay by the *delayed* CSMA with $T = 2$. We further report 'snapshots' of active 'even' and 'odd' links under the delayed CSMA with $T = 1, 2$ in Fig. 3, where the torus graph is bipartite with edges between even and odd nodes (or links). They also explain why $T = 2$ is better than $T = 1$ for the delay performance of the *delayed* CSMA: when $T = 1$ and $T = 2$, the sets of active links change little and much, respectively, between two consecutive times.

VI. Conclusion

In this paper, we have addressed the open question for designing a CSMA algorithm that is both throughput and delay optimal for general wireless network topology. We proved that one of the throughput-optimal CSMA algorithms based on the notion of virtual channels proposed in the literature has the per-link $O(1)$-asymptotic-delay for general wireless network topology if the number of virtual channels has the logarithmic order of the mixing time of the underlying CSMA Markov chain. The significance of our result lies in the proof of the existence of a scheduling policy that achieves optimality in both throughput and delay, operating in a fully distributed manner.

Appendix

Proof of (8)**.** The first term in (8) is bounded by:

$$g(T, Q(t) - 1) = \frac{140M}{\varepsilon^3} \exp\left(-\frac{\varepsilon^2 T}{80}\right) + \frac{18}{\varepsilon^2} \exp\left(-\frac{(Q(t)-1)\varepsilon}{9}\right) \le 1/4 \tag{11}$$

where the last inequality is because $T^*, Q(t)$ are large enough so that $T > T^* = \Omega\left(\frac{1}{\varepsilon^3}\log M\right)$ and $Q(t) > C = \Omega(1/\varepsilon^2)$. On the other hand, for the second term in (8), we observe that

$$\begin{aligned} 3 \sum_{i=1}^{49M^2T^2/\varepsilon^2} g\left(T, Q(t) + \sqrt{i}\right) &\le 3 \sum_{i=1}^{49M^2T^2/\varepsilon^2} \frac{140M}{\varepsilon^3} \exp\left(-\frac{\varepsilon^2 T}{80}\right) \\ &\quad + 3 \sum_{i=1}^{\infty} \frac{18}{\varepsilon^2} \exp\left(-\frac{(Q(t)+\sqrt{i})\varepsilon}{9}\right) \end{aligned}$$

where one can check that the first and second summations in the above inequality can be made arbitrarily small by choosing large $T^*, Q(t)$ with $T > T^* = \Omega\left(\frac{1}{\varepsilon^3}\log M\right)$ and $Q(t) > C = \Omega(1/\varepsilon^2)$, respectively, i.e., $\sum_{i=1}^{49M^2T^2/\varepsilon^2} g\left(T, Q(t) + \sqrt{i}\right) \le \frac{1}{4}$. Combining the above inequality with (11) leads to the proof of (8).

Proof of (10)**.** We first characterize two inequalities that are used to prove (10). For $\zeta > -\varepsilon T/2$ and $i = M+1, M+2, \ldots$, it follows that

$$\begin{aligned} &\mathbb{P}\Big[\Delta_v[t + (i-1)T + 1, t + iT] > \zeta \,|\, \mathcal{L}\Big] \\ &\le \mathbb{P}\Big[Y^v_{t+iT} - Y^v_{t+(i-1)T} \ge \frac{\varepsilon T}{2} + \zeta \,|\, \mathcal{L}\Big] \\ &\le \exp\left(\frac{-(\varepsilon T/2 + \zeta)^2}{2\sum_{i=1}^{T}(1+\varepsilon/2)^2}\right) \\ &\le \exp\left(\frac{-(\varepsilon T/2 + \zeta)^2}{5T}\right), \end{aligned} \tag{12}$$

where we use Azuma's inequality on the supermartingale $\{Y^v_{t+(i-1)T+1}, \ldots Y^v_{t+iT}\}$. Similarly, for $\zeta > 0$, we have that

$$\begin{aligned} &\mathbb{P}\Big[\max_{\substack{(i-1)T+1 \\ < s \le iT}} \Delta_v[t+s, t+iT] \ge \zeta \,|\, \mathcal{L}\Big] \\ &\le \sum_{s=t+(i-1)T+1}^{t+iT} \mathbb{P}\Big[\Delta_v[t+s, t+iT] \ge \zeta \,|\, \mathcal{L}\Big] \\ &= \sum_{\substack{s=t+ \\ (i-1)T+1}}^{t+iT} \mathbb{P}\Big[Y^v_{t+iT} - Y^v_{s-1} \ge \frac{(t+iT-s+1)\varepsilon}{2} + \zeta \,|\, \mathcal{L}\Big] \\ &\le \sum_{s=t+(i-1)T+1}^{t+iT} \exp\left(\frac{-\{(t+iT-s+1)\varepsilon/2 + \zeta\}^2}{2(1+\varepsilon/2)^2(t+iT-s+1)}\right) \end{aligned}$$

$$= \sum_{k=1}^{T} \exp\left(\frac{-\{\frac{\varepsilon k}{2}+\zeta\}^2}{2(1+\frac{\varepsilon}{2})^2 k}\right) \le \sum_{k=1}^{\infty} \exp\left(\frac{-\{\frac{\varepsilon k}{2}+\zeta\}^2}{\frac{4(1+\frac{\varepsilon}{2})^2}{\varepsilon}\{\frac{\varepsilon k}{2}+\zeta\}}\right)$$
$$= \sum_{k=1}^{\infty} \exp\left(\frac{-\frac{\varepsilon^2 k}{2}-\zeta\varepsilon}{4(1+\frac{\varepsilon}{2})^2}\right) \le \sum_{k=1}^{\infty} \exp\left(\frac{-\frac{\varepsilon^2 k}{2}-\zeta\varepsilon}{9}\right)$$
$$= \exp\left(-\frac{\zeta\varepsilon}{9}\right)\cdot\frac{1}{e^{\frac{\varepsilon^2}{18}}-1} \le \frac{18}{\epsilon^2}\exp\left(-\frac{\zeta\varepsilon}{9}\right) \tag{13}$$

Using (12) and (13), we have:

$$\mathbb{P}\Big[\max_{MT+2\le s\le 7MT/\varepsilon} \Delta_v[t+s,t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}\Big]$$
$$= 1-\mathbb{P}\Big[\underbrace{\max_{MT+2\le s\le 7MT/\varepsilon} \Delta_v[t+s,t+7MT/\varepsilon] < \varphi}_{(\star)} \,|\, \mathcal{L}\Big]$$
$$\overset{(a)}{\le} 1-\mathbb{P}\Big[\underbrace{\max_{(i-1)T<s\le iT} \Delta_v[t+s,t+iT] < \varphi+\frac{\varepsilon}{4}T}_{(\star\star)}$$
$$\text{for } \ i=M+1,\dots,7M/\varepsilon-1$$
$$\text{and } \ \underbrace{\max_{(7M/\varepsilon-1)T<s\le 7MT/\varepsilon} \Delta_v[t+s,t+7MT/\varepsilon] < \varphi}_{(\star\star\star)}$$
$$\text{and } \ \underbrace{\Delta_v[t+(i-1)T+1,t+iT] \le -\frac{\varepsilon}{4}T}_{(\star\star\star\star)}$$
$$\text{for } \ i=M+1,\dots,7M/\varepsilon \,|\, \mathcal{L}\Big]$$
$$\le \sum_{i=M+1}^{7M/\varepsilon-1} \mathbb{P}\Big[\max_{(i-1)T<s\le iT} \Delta_v[t+s,t+iT] \ge \varphi+\frac{\varepsilon}{4}T \,|\, \mathcal{L}\Big]$$
$$+\mathbb{P}\Big[\max_{(7M/\varepsilon-1)T<s\le 7MT/\varepsilon} \Delta_v[t+s,t+7MT/\varepsilon] \ge \varphi \,|\, \mathcal{L}\Big]$$
$$+\sum_{i=M+1}^{7M/\varepsilon} \mathbb{P}\Big[\Delta_v[t+(i-1)T+1,t+iT] > -\frac{\varepsilon}{4}T \,|\, \mathcal{L}\Big]$$
$$\le \left(\frac{7}{\varepsilon}M-M-1\right)\frac{18}{\varepsilon^2}\exp\left(\frac{-(\varphi+\frac{\varepsilon}{4}T)\varepsilon}{9}\right)$$
$$+\frac{18}{\varepsilon^2}\exp\left(\frac{-\varphi\varepsilon}{9}\right)+M\left(\frac{7}{\varepsilon}-1\right)\exp\left(\frac{-(\frac{\varepsilon T}{2}-\frac{\varepsilon}{4}T)^2}{5T}\right)$$
$$\le \frac{126M}{\varepsilon^3}\exp\left(\frac{-\varepsilon^2 T}{36}\right)+\frac{18}{\varepsilon^2}\exp\left(\frac{-\varphi\varepsilon}{9}\right)$$
$$+\frac{7}{\varepsilon}M\exp\left(\frac{-\varepsilon^2 T}{80}\right)$$
$$\le \frac{133M}{\varepsilon^3}\exp\left(-\frac{\varepsilon^2 T}{80}\right)+\frac{18}{\varepsilon^2}\exp\left(-\frac{\varphi\varepsilon}{9}\right).$$

In (a), we consider a time interval $[t+MT, t+7MT/\varepsilon]$ that are partitioned into two disjoint sub-intervals $I_1=[t+MT, t+(7M/\varepsilon-1)T]$ and $I_2=[t+(7M/\varepsilon-1)T+1, t+7MT/\varepsilon]$, and we let s^* be the maximization point in $(\star)$. Then, if $s^*\in I_1$, $(\star\star)$ and $(\star\star\star\star)$ imply $(\star)$, and if $s^*\in I_2$, $(\star\star\star)$ is equivalent to $(\star)$. This completes the proof of (10).

REFERENCES

[1] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1949, 1992.

[2] P. Giaccone, B. Prabhakar, and D. Shah, "Randomized Scheduling Algorithms for High-Aggregate Bandwidth Switches," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 546-559, 2003.

[3] E. Modiano, D. Shah, and G. Zussman, "Maximizing Throughput in Wireless Network Via Gossiping," In *Proceedings of ACM SIGMETRICS/Performance*, 2006.

[4] L. Tassiulas, "Linear Complexity Algorithms for Maximum Throughput in Radio Networks and Input Queued Switches," In *Proceedings of IEEE INFOCOM*, vol. 2, pp. 533-539, 1998.

[5] S. Rajagopalan, D. Shah, and J. Shin, "Network Adiabatic Theorem: An Efficient Randomized Protocol for Contention Resolution," In *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, 2009.

[6] D. Shah and J. Shin, "Randomized Scheduling Algorithm for Queueing Networks," *Annals of Applied Probability*, vol. 22, no. 1, pp. 128-171, 2012.

[7] L. Jiang, D. Shah, J. Shin, and J. Walrand, "Distributed Random Access Algorithm: Scheduling and Congestion Control," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6182-6207, 2010.

[8] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Towards utility-optimal random access without message passing," *Wiley Journal of Wireless Communications and Mobile Computing*, vol. 10, no. 1, pp. 115–128, Jan. 2010.

[9] L. Jiang and J. Walrand, "A distributed algorithm for maximal throughput and optimal fairness in wireless networks with a general interference model," EECS Department, University of California, Berkeley, Tech. Rep., Apr 2008. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-38.html

[10] D. Shah and J. Shin, "Delay Optimal Queue-based CSMA," In *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1, 2010.

[11] M. Lotfinezhad and P. Marbach, "Throughput-optimal Random Access with Order-Optimal Delay," In *Proceedings of IEEE INFOCOM*, 2011.

[12] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast Mixing of Parallel Glauber Dynamics and Low-Delay CSMA Scheduling," In *Proceedings of IEEE INFOCOM*, 2011.

[13] V. Subramanian and M. Alanyali, "Delay Performance of CSMA in Networks Bounded Degree Conflict Graphs," In *Proceedings of IEEE ISIT*, 2011.

[14] C.-H. Lee, D. Y. Eun, S.-Y. Yun, and Y. Yi, "From Glauber dynamics to Metropolis algorithm: Smaller delay in optimal CSMA," in *Proceedings of ISIT*, June 2012.

[15] 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 *Proceedings of IEEE ISIT*, 2012.

[16] P.-K. Huang and X. Lin, "Improving the Delay Performance of CSMA Algorithms: A Virtual Multi-Channel Approach," In *Proceedings of IEEE INFOCOM*, 2013.

[17] J. Kwak, C.-H. Lee and D. Y. Eun, "Exploiting the Past to Reduce Delay in CSMA Scheduling: A High-order Markov Chain Approach," in *Proceedings of ACM SIGMETRICS*, pp. 657-668, 2013.

[18] M. J. Neely, "Delay Analysis for Maximal Scheduling in Wireless Networks with Bursty Traffic," in *Proceedings of IEEE INFOCOM*, 2008.

[19] Y. Yi and M. Chiang, "Complexity in Wireless Scheduling," in *Proceedings of ACM MOBIHOC*, 2008.

[20] M. J. Neely, E. Modiano, and Y. S. Cheng. "Logarithmic delay for N N packet switches under the crossbar constraint," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 657-668, 2007.

[21] D. Shah, N. Walton, and Y. Zhong. "Optimal queue-size scaling in switched networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, 2012.

[22] D. Shah, D. N. C. Tse and J. N. Tsitsiklis, "Hardness of low delay network scheduling," *IEEE Transactions on Information Theory*, vol. 57, no. 12, 2011.

[23] N. Hegde and A. Proutiere, "Simulation-based optimization algorithms with applications to dynamic spectrum access," in *Proceedings of CISS*, 2012.

[24] N. Berger, C. Kenyon, E. Mossel, and Y. Peres. "Glauber dynamics on trees and hyperbolic graphs," *Probability Theory and Related Fields*, vol. 131, no. 3, pp. 311–340, 2005.

[25] J. Ni, B. Tan, and R. Srikant, "Q-csma: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," in *Proceedings of IEEE INFOCOM*, 2010.