Optimal Distributed Scheduling under Time-varying Conditions: A Fast-CSMA Algorithm with Applications

Bin Li and Atilla Eryilmaz

Abstract—**Recently, low-complexity and distributed Carrier Sense Multiple Access (CSMA)-based scheduling algorithms have attracted extensive interest due to their throughput-optimal characteristics in general network topologies. However, these algorithms are not well-suited for time-varying environments (i.e., serving real-time traffic under time-varying channel conditions in wireless networks) for two reasons: (1) the mixing time of the underlying CSMA Markov Chain grows with the size of the network, which, for large networks, generates unacceptable delay for deadline-constrained traffic; (2) since the dynamic CSMA parameters are influenced by the arrival and channel state processes, the underlying CSMA Markov Chain may not converge to a steady-state under strict deadline constraints and fading channel conditions.**

In this paper, we attack the problem of distributed scheduling for time-varying environments. Specifically, we propose a Fast-CSMA (FCSMA) policy in fully-connected topologies, which converges much faster than the existing CSMA algorithms and thus yields significant advantages for time-varying applications. Then, we design optimal policies based on FCSMA techniques in two challenging and important scenarios in wireless networks for scheduling inelastic traffic with/without channel state information (CSI) over wireless fading channels.

I. Introduction

Efficient utilization of network resources requires careful interference management among simultaneous transmissions. Of particular interest in the efficient scheduling are Queue-Length-Based (QLB) schedulers (e.g., [2], [3], [4], [5], [6]) due to their provably optimal performance guarantees. Randomization is useful in allowing flexibilities in the design and implementation of such schedulers (e.g., [7]). However, it causes inaccurate operation and may be hurtful if it is not performed within limits (see [8] for more details). One of the most robust randomized schedulers is CSMA-based distributed scheduler (e.g., [9], [10], [11], [12]), whose stationary distribution of the underlying Markov chain has a product-form.

It is well-known that CSMA-based scheduler can maximize long-term average throughput for general wireless topologies. However, these results do not apply to time-varying environments (i.e., scheduling deadline-constrained traffic over wireless fading channels), since their throughput-optimality relies: (i) on the convergence time of the underlying Markov Chain to its steady-state, which grows with the size of the network; and (ii) on relatively stationary conditions in which the CSMA parameters do not change significantly over time so that the instantaneous service rate distribution can stay close to the stationary distribution. Both of these conditions are violated in time-varying environments. For example, packets of deadline-constrained traffic are likely to be dropped before the CSMA-based algorithm converges to its steady-state, and the time-varying fading creates significant variations on the CSMA parameters, in which case the instantaneous service rate distribution cannot closely track the stationary distribution. To the best of our knowledge, there does not exist a work that can achieve provably good performance by using attractive CSMA principles under time-varying conditions.

While achieving low delay via distributed scheduling in general topologies is a difficult task (see [13]), in a related work [14] that focuses on grid topologies, the authors have designed an Unlocking CSMA (UCSMA) algorithm with both maximum throughput and order optimal average delay performance, which shows promise for low-delay distributed scheduling in special topologies. However, UCSMA also does not directly apply to deadline-constrained traffic since its measure of delay is on average. Moveover, it is not clear how existing CSMA or UCSMA will perform under fading channel conditions. Thus, designing an optimal distributed scheduling algorithm in time-varying environments remains an open question.

With this motivation, in this work, we address the problem of distributed scheduling in fully connected networks (e.g., Cellular network, Wi-Fi network) for time-varying environments. We propose a Fast-CSMA (FCSMA) algorithm that, despite its similarity of name, fundamentally differs from existing CSMA policies in its design principle: rather than evolving over the set of schedules to reach a favorable steady-state distribution, the FCSMA policy aims to quickly reach one of a set of favorable schedules and stick to it for a duration related to time-varying scale of the application. While the performance of the former strategy is tied to the mixing-time of a Markov Chain, the performance of our strategy is tied to the hitting time, and hence, yields significant advantage for time-varying applications.

In this work, we apply FCSMA techniques in two main scenarios: deadline-constrained scheduling with/without channel state information (CSI) over wireless fading channels. We also consider the application of FCSMA techniques in non-deadline-constrained scheduling over wireless fading channels in our technical report [15]. The two scenarios we considered

This work is supported by the NSF grants: CAREER-CNS-0953515 and CCF-0916664. Also, the work of A. Eryilmaz was in part supported by the QNRF grant number NPRP 09-1168-2-455

An earlier version of this paper has appeared in the workshop on Resource Allocation and Cooperation in Wireless Networks (RAWNET), Princeton, NJ, May, 2011 [1].

Bin Li and Atilla Eryilmaz ({lib,eryilmaz}@ece.osu.edu) are with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, Ohio 43210 USA.

in this paper are most challenging and important application in practice, since wireless networks are expected to serve real-time traffic, such as video or voice applications, generated by a large number of users over potentially fading channels. These constraints and requirements, together with the limited shared resources, generate a strong need for distributed algorithms that can efficiently utilize the available resources while maintaining high quality-of-service for the real-time applications. Yet, the strict short-term deadline constraints and long-term throughput requirements associated with most real-time applications complicate the development of provably good distributed solutions.

All existing works in deadline-constrained scheduling (e.g., [16], [17], [18], [19]) assume centralized controllers, and hence are not suitable for distributed operation. To the best of our knowledge, this is the first work that proposes an optimal and distributed algorithm under time-varying conditions caused by channel fading or time-sensitive applications. Our main contributions in this paper are:

• In Section II, we propose a FCSMA algorithm that aims to quickly reach one of a set of favorable schedules.

• We design an optimal distributed policy based on FC-SMA techniques in scheduling deadline-constrained traffic with/without CSI over wireless fading channels in Section III and Section IV, respectively.

II. The principle of Fast-CSMA design

We consider a fully-connected network topology where L links contend for data transmission over a single channel. We assume a time-slotted system, where all links start transmission at the beginning of each time slot. Due to the interference constraints, at most one link can transmit in each slot. We call a schedule where at most one link is active in each slot as a *feasible schedule*.

Randomized schedulers (e.g., [7], [9], [20], [21], [22] and [23]) are widely studied due to their flexibilities in development of low-complexity and distributed implementations. The most promising and interesting randomized schedulers are distributed CSMA-based algorithms. We give the definition of continuous-time CSMA algorithm (see [9]) for completeness. In this paper, we adopt the same assumptions as in [9] that the sensing is instantaneous and the backoff time is continuous.

Definition 1 (CSMA Algorithm): Each link l independently generates an exponentially distributed random variable with rate $R_l[t]$ and starts transmitting after this random duration unless it senses another transmission before. If link l senses the transmission, it suspends its backoff timer and resumes it after the completion of this transmission. The transmission time of each link is exponentially distributed with mean 1.

Figure 1a shows the state transition diagram of the underlying Markov Chain for the CSMA Algorithm when there are 3 available links at time t, where each state stands for a feasible schedule. It is easy to see that the stationary distribution of this Markov Chain is

$$P_l = \frac{R_l[t]}{1+Z[t]}, \forall l, \tag{1}$$

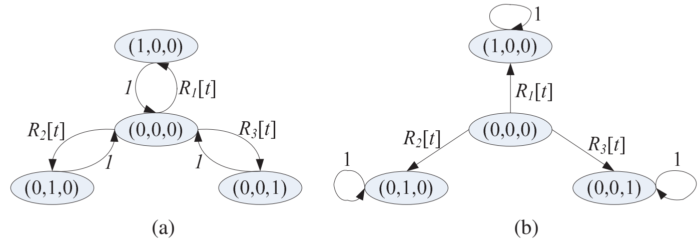


Fig. 1: (a) Markov chain for a CSMA algorithm (b) Markov chain for a FCSMA algorithm

where $Z[t] = \sum_{l=1}^{L} R_l[t]$. Since $\mathbf{R}[t] = (R_l[t])_{l=1}^{L}$ is chosen as a function of network state information (e.g., queue length, channel state information, arrivals) in wireless networks, the underlying Markov Chain for the CSMA Algorithm is inhomogeneous. Intuitively, the CSMA parameters $\mathbf{R}[t]$ should change slowly such that the instantaneous service rate distribution can stay close to the stationary distribution. Indeed, for the application of scheduling over time-invariant channels (i.e., the transmission rate of each link does not change over time), such mapping has been observed to be optimal (e.g., [11] and [12]) if the CSMA parameter $R_l[t]$ of each link l can take certain functional forms (e.g., $\log\log(\cdot)$) of its queue length at time t. Note that the queue length will change slowly when it is large enough. The purpose of choosing the slowly increasing function is further to make the CSMA parameters as a function of queue length do not change significantly over time.

However, for the application of scheduling over wireless fading channels, the CSMA parameters $\mathbf{R}[t]$ need to be chosen as a function of channel sate information to yield good performance. In such case, no matter what function we choose for the channel state, $\mathbf{R}[t]$ will change significantly as the fading state fluctuates and thus the instantaneous service distribution is not expected to track the stationary distribution. More generally, extending CSMA solutions to stochastic network dynamics or sophisticated application requirements (e.g., serving traffic with strict deadline constraints over wireless fading channels) is difficult for two reasons: 1) the mixing time of the underlying CSMA Markov chain grows with the size of the network, which, for large networks, generates unacceptable delay for deadline-constrained traffic; 2) since the dynamic CSMA parameters $\mathbf{R}[t]$ are influenced by the arrival and channel state process, the underlying CSMA Markov chain may not converge to its steady-state under strict deadline constraints and wireless fading channel conditions.

Thus, designing an optimal and distributed scheduling algorithm for stochastic networks becomes quite challenging. In this paper, we propose a Fast-CSMA strategy that provides provably good performance under time varying conditions. Our approach fundamentally differs from existing CSMA solutions in that our FCSMA policy exploits the fast convergence characteristics of ”hitting times” instead of ”mixing times”.

Definition 2 (Fast-CSMA (FCSMA) Algorithm): At the beginning of each time slot t, each link l independently generates an exponentially distributed random variable with rate $R_l[t]$, and starts transmitting after this random duration unless it senses another transmission before. If all links have their random duration greater than a slot, all links will keep silent in the current slot; otherwise, the link that captures the channel transmits its data[1] until the end of the slot. The whole process is repeated in the next time slot.

Remarks: (1) The operation of the FCSMA Algorithm resembles that of the UCSMA Algorithm (see [14]). The difference lies in that the UCSMA algorithm restarts the CSMA Algorithm to achieve both maximum throughput and order optimal average delay in grid network topologies over time-invariant channels by carefully choosing the running period. However, it is unclear whether the UCSMA algorithm can still work well in time-varying applications.

(2) By choosing the running period for the FCSMA Algorithm the same as the time scale of network dynamics (i.e., the block length for block fading or maximum allowable deadline), we can show in later sections that the FCSMA Algorithm exhibits excellent performance in time-varying applications.

(3) In general multi-hop network topologies, the FCSMA Algorithm can still converge very fast to one feasible schedule. Yet, the probability of serving each schedule may not have a product form and the performance of the FCSMA Algorithm is unclear. We leave it for future investigation.

Figure 1b gives the state transition diagram of underlying Markov Chain for the FCSMA Algorithm when there are 3 available links, where each state represents a feasible schedule. The convergence time of the FCSMA Algorithm is tied to the hitting time[2], while the convergence time of the CSMA Algorithm is dominated by the mixing time of Markov chain, which generally is large. The hitting time of the FCSMA Algorithm at slot t is exponentially distributed with mean $\frac{1}{Z[t]}$, which is generally small in practice as we will see in simulations. Due to its small hitting time, the FCSMA Algorithm yields significant advantages over existing CSMA policies evolving slowly to the steady-state and may work well in more challenging environments, i.e., scheduling real-time traffic over wireless fading channels. Because of the fast convergence property of the FCSMA Algorithm, we introduce the idealized FCSMA algorithm for easier theoretical analysis. The simulation results in the later sections indicate that both FCSMA and Idealized FCSMA Algorithm have the same system performance.

Definition 3 (Idealized FCSMA Algorithm): Idealized FCSMA Algorithm is the FCSMA Algorithm with zero hitting time, which assumes that it can reach the favorable state instantaneously.

For the Idealized FCSMA Algorithm, the probability of serving the link l in each slot t will be:

$$\pi_l[t] = \frac{R_l[t]}{Z[t]}. \tag{2}$$

Let $W_l[t] = \log(R_l[t])$ and $W^*[t] = \max_l W_l[t]$. The following lemma establishes the fact that the Idealized FCSMA Algorithm picks a link with the weight close to the maximum weight with high probability when the maximum weight $W^*[t]$ is large enough at each slot t.

Lemma 1: Given $\epsilon > 0$ and $\zeta > 0$, $\exists \overline{W} \in (0, \infty)$, such that if $W^*[t] > \overline{W}$, then the Idealized FCSMA Algorithm picks a link l satisfying

$$\Pr\{W_l[t] \geq (1-\epsilon)W^*[t]\} \geq 1 - \zeta. \tag{3}$$

The proof is similar to that in [10] and [8], and thus is omitted here for brevity.

In the rest of paper, we mainly consider inelastic traffic. The inelastic traffic means that each arrival has a maximum delay requirement while the elastic traffic does not have such a requirement. We apply the FCSMA technique in two challenging scenarios: scheduling inelastic traffic with/without Channel State Information (CSI) over wireless fading channels. In our technical report [15], we also consider scheduling elastic traffic over wireless fading channels. In each application, we need to carefully design the FCSMA parameters $\mathbf{R}[t] = (R_l[t])_{l=1}^L$ at each slot t to yield optimal performance. To facilitate the flexibility in the design and implementation of the FCSMA algorithm, we define a set of functions (also see [8]):

$\mathcal{F} \triangleq$ set of non-negative, nondecreasing and differentiable functions $f(\cdot): \mathbb{R}_+ \to \mathbb{R}_+$ with $\lim_{x\to\infty} f(x) = \infty$.

$\mathcal{B} \triangleq \{f \in \mathcal{F}\text{: } \lim_{x\to\infty} \frac{f(x+a)}{f(x)} = 1, \text{ for any } a \in \mathbb{R}\}$.

The examples of functions that are in class $\mathcal{B}$ are $f(x) = \log x$, $f(x) = x$ or $f(x) = e^{\sqrt{x}}$. Note that $f(x) = e^x$ does not belong to class $\mathcal{B}$.

Now, we are ready to develop optimal FCSMA algorithms in two challenging applications: scheduling inelastic traffic with/without CSI over wireless fading channels.

III. Scheduling Inelastic Traffic with CSI

A. Basic Setup

We assume that the wireless channel is independently block fading at each link. We capture the channel fading over link l via $C_l[t]$, which measures the maximum amount of service available in slot t, if scheduled. We assume that $\mathbf{C}[t] = (C_l[t])_{l=1}^L$ are independently distributed random variables over links and identically distributed over time with $C_l[t] \leq C_{\max}, \forall l, t$, for some $C_{\max} < \infty$. We use a binary variable $S_l[t]$ to denote whether the link l is served at slot t, where $S_l[t] = 1$ if the link l can be served at slot t and $S_l[t] = 0$, otherwise. Let $\mathbf{S}[t] = (S_l[t])_{l=1}^L$ be a feasible schedule. We use $\mathcal{S}$ to denote the collection of feasible schedules. Recall that at most one link can be active in a feasible schedule, due to the fully connected network topology we consider in this paper.

[1] If there is no data awaiting in the link l, it transmits dummy data to occupy the channel.

[2] The hitting time is an empty duration after which the Markov Chain stays in a non-zero feasible schedule state (i.e., the channel is occupied by one of users)

We assume that all arrivals have the same delay bound of T time slots, which means that if the data cannot be served during T slots after it arrives, it will be dropped. For convenience, we call a set of T consecutive time slots a *frame*. In the context of fully-connected networks, we associate each real-time flow with a link, and hence use these two terms interchangeably. We assume that all data arrives at each link at the beginning of each frame. Let $A_l[kT]$ denote the amount of data arriving at link l in frame k that are independently distributed over links and identically distributed over time with mean λ_l, and $A_l[kT] \leq A_{\max}$ for some $A_{\max} < \infty$. All the remaining data is dropped at the end of a frame. Each link has a maximum allowable drop rate $\rho_l \lambda_l$, where $\rho_l \in (0, 1)$ is the maximum fraction of data that can be dropped at link l. For example, $\rho_l = 0.1$ means that at most 10% of data can be dropped at link l on average.

Our goal is to find the schedule $\{\mathbf{S}[t]\}_{t\geq 1}$ under the scheduling constraint that at most one link can be scheduled at each time slot and dropping rate constraint that the average drop rate of each link should not be greater than its maximum allowable drop rate. To solve this optimal control problem, we use the intelligent technique in [24] to introduce a virtual queue $X_l[kT]$ for each link l to track the amount of dropped data in frame k. Specifically, the amount of data arriving at virtual queue l at the end of frame k is denoted as $D_l[kT]$, which is equal to $A_l[kT] - \min\left\{\sum_{t=kT}^{(k+1)T-1} C_l[t]S_l[t], A_l[kT]\right\}$. We use $I_l[kT]$ to denote the service for virtual queue l at the end of the frame k with mean $\rho_l \lambda_l$, and $I_l[kT] \leq I_{\max}$ for some $I_{\max} < \infty$. Further, we let $U_l[kT]$ denote the unused service for queue l at the end of frame k, which is upper-bounded by $I_{\max}$. Then, the evolution of virtual queue l is described as follows:

$$X_l[(k+1)T] = X_l[kT] + D_l[kT] - I_l[kT] + U_l[kT], \forall l. \quad (4)$$

In this and next section, we consider two main scenarios: known channel state and unknown channel state. For the known channel state case, we assume that the channel state is constant for the duration of a frame and each link knows CSI at the beginning of each frame. For the unknown channel state case, we allow that the channel state changes from time slot to time slot and each link does not know CSI before each transmission, but can determine how much data has been transmitted at each slot after we get feedback from the receiver. These assumptions are also adopted in [18].

We consider the class of stationary policies $\mathcal{P}$ that select $\mathbf{S}[t]$ as a function of $(\mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT])$ for the known channel state scenario and a function of $(\mathbf{X}[kT], \mathbf{A}[kT])$ for the unknown channel state scenario in frame k, which, then, form a Markov Chain, where $\mathbf{X}[kT] = (X_l[kT])_{l=1}^L$ and $\mathbf{A}[kT] = (A_l[kT])_{l=1}^L$. If this Markov Chain is positive recurrent, then the average drop rate will meet the required dropping rate constraint automatically (see [25]). We define the maximal satisfiable region as a maximum set of arrival processes for which this Markov Chain is positive recurrent under any policy. We call an algorithm *optimal* if it makes Markov Chain positive recurrent for any arrival process within the maximal satisfiable region.

B. FCSMA algorithm implementation

In this subsection, we first characterize the maximal satisfiable region and then propose an optimal FCSMA algorithm with CSI for scheduling inelastic traffic over fading channels.

Consider the class $\mathcal{P}$ of stationary policies that base their scheduling decision on the observed vector $(\mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT])$ in frame k. The next lemma establishes a necessary condition for stabilizing the system.

Lemma 2: If there is a policy $\mathbf{P}_0 \in \mathcal{P}$ that can stabilize the virtual queue $\mathbf{X}$, then there exist non-negative numbers $\alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1})$ such that

$$\sum_{\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1} \in \mathcal{S}} \alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1}) = 1, \forall \mathbf{a}, \mathbf{c}, \quad (5)$$

$$\lambda_l(1-\rho_l) < \sum_{\mathbf{a}} P_{\mathbf{A}}(\mathbf{a}) \sum_{\mathbf{c}} P_{\mathbf{C}}(\mathbf{c}) \sum_{\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1} \in \mathcal{S}} \alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1}) \min\left\{\sum_{i=0}^{T-1} c_l s_l^i, a_l\right\}, \forall l, \quad (6)$$

where $\mathbf{s}^i = (s_l^i)_{l=1}^L$, $P_{\mathbf{A}}(\mathbf{a}) = P(\mathbf{A}[t] = \mathbf{a})$ and $P_{\mathbf{C}}(\mathbf{c}) = P(\mathbf{C}[t] = \mathbf{c})$.

The proof is almost the same as in [26]. The main difference lies in that our proof deals with the necessary condition for stabilizing virtual queues instead of data queues as in [26]. We omit it for conciseness. Note that the right hand side of the inequality (6) is the average service provided for link l during one frame; while $\lambda_l(1-\rho_l)$ is the average amount of data at link l that needs to be served. Thus, to meet the maximum allowable drop rate requirement, (6) should be satisfied. We define the *maximal satisfiable region* $\Lambda_1(\boldsymbol{\rho}, \mathbf{C})$ as follows:

$$\Lambda_1(\boldsymbol{\rho}, \mathbf{C}) \triangleq \{\mathbf{A} : \exists \alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1}) \geq 0, \text{ such that both (5) and (6) satisfy}\}.$$

We are now ready to develop an optimal centralized algorithm with CSI for scheduling inelastic traffic over wireless fading channels.

Centralized Algorithm with CSI:

In each frame k, given ($\mathbf{X}[kT]$,$\mathbf{A}[kT]$,$\mathbf{C}[kT]$), perform

$$\{\mathbf{S}^*[t]\}_{t=kT}^{(k+1)T-1} \in \underset{\{\mathbf{S}[t]\}_{t=kT}^{(k+1)T-1}}{\text{argmax}} \sum_l f(X_l[kT]) \min\left\{C_l[kT] \sum_{t=kT}^{(k+1)T-1} S_l[t], A_l[kT]\right\}, \quad (7)$$

where $f \in \mathcal{F}$.

Remark: In [18], the authors proposed a centralized algorithm with $f(x) = x$. Our proposed centralized algorithm is more general, which allows more flexibilities in distributed implementations.

Next, we establish the optimality of the centralized algorithm with CSI under certain conditions for function f.

Theorem 1: If $f \in \mathcal{B}$, the Centralized Algorithm with CSI for scheduling inelastic traffic is optimal over wireless fading channels, i.e., for any arrival process $\mathbf{A} \in \Lambda_1(\boldsymbol{\rho}, \mathbf{C})$, it makes the underlying Markov Chain positive recurrent.

The proof is a generalization of that in [18] and is a special case of that in Theorem 3, where we use FCSMA techniques to mimic the Centralized Algorithm. Thus, we omit it for brevity. Even though the above centralized algorithm is optimal, it cannot directly be applied in practice due to the need of centralized coordination. Next, we propose a greedy algorithm that is well suited for distributed implementation. To that end, we first give the key identity that facilitates the development of greedy solutions.

Lemma 3: Let $a \geq 0$ and $c[t] \geq 0, \forall t = 0, 1, ..., T-1$. If $s[t] \in \{0, 1\}, \forall t$, then

$$\min\left\{\sum_{t=0}^{T-1} c[t]s[t], a\right\} = \sum_{t=0}^{T-1} \min\left\{c[t], \left(a - \sum_{j=0}^{t-1} c[j]s[j]\right)^+\right\} s[t], \quad (8)$$

where $(x)^+ = \max\{x, 0\}$.

Proof: The proof directly follows by induction. Please see our technical report [15] for details. ■

Based on Lemma 3, the objective function in (7) can be rewritten as

$$\sum_l f(X_l[kT]) \min\left\{C_l[kT] \sum_{t=kT}^{(k+1)T-1} S_l[t], A_l[kT]\right\}$$
$$= \sum_{t=kT}^{(k+1)T-1} \sum_l f(X_l[kT]) \min\left\{C_l[kT], \left(A_l[kT] - C_l[kT] \sum_{j=kT}^{t-1} S_l[j]\right)^+\right\} S_l[t]. \quad (9)$$

We can observe that the equation (9) decouples the scheduling decisions over a frame and help develop the greedy solutions that are easy to be implemented distributively.

Greedy Algorithm with CSI:

At each time slot $t \in \{kT, kT+1, ..., (k+1)T-1\}$ in frame k, select link $l^G[t]$ such that

$$l^G[t] \in \underset{l}{\operatorname{argmax}}\, f(X_l[kT]) \min\left\{C_l[kT], \left(A_l[kT] - C_l[kT] \sum_{j=kT}^{t-1} S_l[j]\right)^+\right\}, \quad (10)$$

where $f \in \mathcal{F}$.

Theorem 2: The Greedy Algorithm with CSI is an optimal solution to the problem (7) and thus is optimal for scheduling inelastic traffic over wireless fading channels if $f \in \mathcal{B}$.

The proof is a special case of that in Theorem 5: the channel state is constant over a frame in the proof for Theorem 2, while the channel state changes from slot to slot in a frame in that for Theorem 5, which makes it more challenging to deal with. Next, we expand on the distributed implementation of the greedy solution by using the FCSMA technique.

Idealized FCSMA Algorithm with CSI:

At each time slot $t \in \{kT, kT+1, ..., (k+1)T-1\}$ in frame k, choose the rates

$$R_l[t] = g(X_l[kT])^{\min\left\{C_l[kT], \left(A_l[kT] - C_l[kT]\sum_{j=kT}^{t-1} S_l[j]\right)^+\right\}}, \forall l, \quad (11)$$

where $g \in \mathcal{F}$.

Note that the Idealized FCSMA Algorithm with CSI does not take the convergence time into consideration. For the FCSMA Algorithm with CSI, the rates can be selected as

$$R_l[t] = g(X_l[kT])^{\min\{C_l[kT], J_l[t]\}},$$

for any link l and any $t \in \{kT, kT+1, ..., (k+1)T-1\}$, where $J_l[t]$ is the remaining data at link l at the beginning of each time slot t. Next, we will show that the Idealized FCSMA Algorithm yields the optimal performance. Simulation results show that both FCSMA and Idealized FCSMA Algorithm have the same performance.

Theorem 3: If $f(x) = \log g(x) \in \mathcal{B}$ and $g(0) \geq 1$, the Idealized FCSMA Algorithm with CSI for scheduling inelastic traffic is optimal over wireless fading channels, i.e., for any arrival process $\mathbf{A} \in \Lambda_1(\boldsymbol{\rho}, \mathbf{C})$, it makes the underlying Markov Chain positive recurrent.

Proof: The proof follows from the Lyapunov drift argument. However, it is quite challenging to argue that the Idealized FCSMA Algorithm with CSI mimics the Centralized/Greedy Algorithm with CSI over a frame, which is an obvious case when $T = 1$ (see [1]). By properly partitioning the space of weights chosen by the Greedy Algorithm with CSI within a frame, we tackle this difficulty and refer the reader to see Appendix A for the details. ■

C. Simulation Results

In this subsection, we perform simulations to validate the optimality of the proposed FCSMA policy with CSI for scheduling inelastic traffic with deadline constraint of T slots over wireless fading channels. In the simulation, there are $L = 10$ links and each frame has $T = 5$ slots. All links require the maximum fraction of dropped data to not exceed $\rho = 0.3$. The amount of arrivals in each frame follows common Bernoulli distribution that the amount of arrivals equal to T with probability λ. All links suffer from the ON-OFF channel fading independently with probability $p = 0.8$ that the channel is available in each frame. The service for virtual queue also follows Bernoulli distribution that the maximum available service equals to T with probability $\rho\lambda$. Under this setup, we can use the same technique in paper [3] to get the maximal satisfiable region: $\Lambda_1(\boldsymbol{\rho}, \mathbf{C}) = \{\lambda : L(1-\rho)\lambda < 1-(1-p\lambda)^L\}$. Through numerical calculations, we can get the maximal satisfiable region: $\{\lambda : \lambda < 0.038\}$. In the simulations, we also compare our proposed FCSMA policy with the QCSMA algorithm (see [10]) with the $\log\log$ function.

From Figure 2, we can observe that the FCSMA Algorithm with both $g(x) = e^x$ and $g(x) = x + 1$ can achieve maximal satisfiable region. Also, we see that the average virtual queue length of the FCSMA Algorithm with exponential function is smaller than that with linear function. However, the meaning of smaller virtual queue length is unclear in this setup. We will explore it in our future research. In addition, we can observe that the FCSMA Algorithm has almost the same performance as that with the Idealized FCSMA Algorithm, which indicates that the hitting time should be negligibly small. Furthermore, the QCSMA algorithm with $\log\log$ function cannot even support the arrival rate of $\lambda = 0.001$ (i.e., its corresponding virtual queues are unstable for the arrival rate of 0.001). The reason for the poor performance of the QCSMA algorithm is that it does not have enough time to converge to the steady state under fast dynamics of the arrival and channel processes.

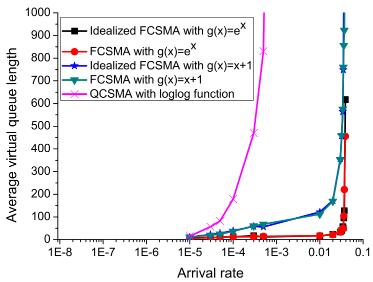


Fig. 2: The performance of the FCSMA Algorithm with CSI

IV. Scheduling Inelastic Traffic without CSI

In this section, we consider the inelastic traffic scheduling without CSI over wireless fading channels. We assume that each link knows how much data has been transmitted at the end of each slot by using per-slot feedback information. The per-slot feedback complicates the design of distributed scheduling algorithm. But, we still can find a similar greedy solution as in Section III and design its distributed algorithm by using FCSMA techniques.

A. FCSMA algorithm implementation

Consider the class $\mathcal{P}$ of stationary policies that base their scheduling decision on the observed vector $(\mathbf{X}[kT], \mathbf{A}[kT])$ in frame k. The next lemma establishes a condition that is necessary for stabilizing the system.

Lemma 4: If there is a policy $\mathbf{P}_0 \in \mathcal{P}$ that can stabilize the virtual queue $\mathbf{X}$, then there exist non-negative numbers $\alpha_0(\mathbf{a};\mathbf{s}^0), \alpha_1(\mathbf{a},\mathbf{s}^0;\mathbf{s}^1), ..., \alpha_{T-1}(\mathbf{a},\mathbf{s}^0,...,\mathbf{s}^{T-2};\mathbf{s}^{T-1})$, such that

$$\sum_{\mathbf{s}^0\in\mathcal{S}} \alpha_0(\mathbf{a};\mathbf{s}^0) = 1, \forall \mathbf{a}, \quad (12)$$

$$\sum_{\mathbf{s}^i\in\mathcal{S}} \alpha_i(\mathbf{a},\mathbf{s}^0,...,\mathbf{s}^{i-1};\mathbf{s}^i) = 1, \forall \mathbf{a}, i = 1,2,...,T-1, \quad (13)$$

$$\lambda_l(1-\rho_l) < \sum_{\mathbf{a}} P_{\mathbf{A}}(\mathbf{a}) \sum_{\mathbf{s}^0,\mathbf{s}^1,...,\mathbf{s}^{T-1}\in\mathcal{S}} \alpha_0(\mathbf{a};\mathbf{s}^0)\alpha_1(\mathbf{a},\mathbf{s}^0;\mathbf{s}^1)... \alpha_{T-1}(\mathbf{a},\mathbf{s}^0,...,\mathbf{s}^{T-2};\mathbf{s}^{T-1})\mathbb{E}\left[\min\left\{\sum_{i=0}^{T-1} c_l s_l^i, a_l\right\}\right], \forall l. \quad (14)$$

The proof is almost the same as [26] and hence is omitted here. We define maximal satisfiable region $\mathbf{\Lambda}_2(\boldsymbol{\rho},\mathbf{C})$ as follows:

$$\mathbf{\Lambda}_2(\boldsymbol{\rho},\mathbf{C}) \triangleq \{\mathbf{A} : \exists \alpha_0(\mathbf{a};\mathbf{s}^0), \alpha_1(\mathbf{a},\mathbf{s}^0;\mathbf{s}^1), ..., \alpha_{T-1}(\mathbf{a},\mathbf{s}^0,...,\mathbf{s}^{T-2};\mathbf{s}^{T-1}) \geq 0, \text{such that both } (12), (13), \text{ and } (14) \text{ satisfy}\}.$$

Next, we develop an optimal centralized algorithm without CSI for scheduling inelastic traffic over fading channels.

Centralized Algorithm without CSI:

In each frame k, given $(\mathbf{X}[kT], \mathbf{A}[kT])$, solve the following optimization problem:

$$\max_{\{\mathbf{S}[t]\}_{t=kT}^{(k+1)T-1}} \mathbb{E}\left[\sum_l f(X_l[kT]) \min\left\{\sum_{t=kT}^{(k+1)T-1} C_l[t]S_l[t], A_l[kT]\right\}\right], \quad (15)$$

where $f \in \mathcal{F}$, and the schedule at each slot is determined after knowing how much data has been transmitted in the previous slots in each frame.

Remark: In [18], the authors designed a centralized algorithm with $f(x) = x$. Our proposed centralized algorithm generalizes this to a large space of functions f, and allows for more flexibilities in distributed implementations.

Next, we establish the optimality of the centralized algorithm without CSI under certain conditions for function f.

Theorem 4: If $f \in \mathcal{B}$, the Centralized Algorithm without CSI for scheduling inelastic traffic is optimal over wireless fading channels, i.e., for any arrival process $\mathbf{A} \in \mathbf{\Lambda}_2(\boldsymbol{\rho},\mathbf{C})$, it makes the underlying Markov Chain positive recurrent.

The proof is a generalization of that in [18] and follows the same argument as that in Theorem 3. Thus, we omit it here for brevity. The centralized algorithm without CSI is quite complicated, since it couples the scheduling decisions in each frame. Under the per-slot feedback assumption, the optimization problem (15) can be solved by using dynamic programming. Based on Lemma 3, we have the following key identity:

$$\sum_l f(X_l[kT]) \min\left\{\sum_{t=kT}^{(k+1)T-1} C_l[t]S_l[t], A_l[kT]\right\} = \sum_{t=kT}^{(k+1)T-1} \sum_l f(X_l[kT]) \min\left\{C_l[t], \left(A_l[kT] - \sum_{j=kT}^{t-1} C_l[j]S_l[j]\right)^+\right\} S_l[t]. \quad (16)$$

By using (16), we can get the following backward equation (see [27]) for the optimization problem (15).

Backward Equation for (15):

At each slot $t \in \{kT, kT+1, ..., (k+1)T-1\}$ in frame k, given $(\mathbf{X}[kT], \mathbf{A}[kT])$ and $\{(\mathbf{C}[j], \mathbf{S}[j])\}_{j=kT}^{t-1}$, select link $l^*[t]$ such that

$$l^*[t] \in \operatorname*{argmax}_l \Bigg(f(X_l[kT])\mathbb{E}\left[\min\left\{C_l[t], \left(A_l[kT] - \sum_{j=kT}^{t-1} S_l[j]C_l[j]\right)^+\right\}\right]$$
$$+ \max_{\{\mathbf{S}[r]\}_{r=t+1}^{(k+1)T-1}} \sum_{i=t+1}^{(k+1)T-1} \sum_{l'=1}^{L} f(X_{l'}[kT])$$
$$\mathbb{E}\left[\min\left\{C_{l'}[i], \left(A_{l'}[kT] - \sum_{j=kT}^{i-1} S_{l'}[j]C_{l'}[j]\right)^+\right\}\right] S_{l'}[i] \Bigg), \quad (17)$$

where $f \in \mathcal{F}$.

At first glance, the optimal solution to problem (17) at each time slot depends on the future slots and thus is difficult to be implemented distributively. However, it may still be possible to decouple the scheduling decisions over a frame, since the channel states are i.i.d. across over time slots. Next, we will show that this is the case in our setup.

Greedy Algorithm without CSI:

At each time slot $t \in \{kT, kT+1, ..., (k+1)T-1\}$ in frame k, given $(\mathbf{X}[kT], \mathbf{A}[kT])$ and $\{(\mathbf{C}[j], \mathbf{S}[j])\}_{j=kT}^{t-1}$, select link $l^G[t]$ such that

$$l^G[t] \in \arg\max_l f(X_l[kT])\mathbb{E}\left[\min\left\{C_l[t], \left(A_l[kT] - \sum_{j=kT}^{t-1} C_l[j]S_l[j]\right)^+\right\}\right], \quad (18)$$

where $f \in \mathcal{F}$.

Theorem 5: The Greedy Algorithm without CSI is optimal for problem (17) and thus is optimal for scheduling inelastic traffic over wireless fading channels if $f \in \mathcal{B}$.

Proof: Without loss of generality, we consider the frame $k = 0$. We will show that if $l^G[t]$ satisfies (18) at time $t \in \{0, 1, ..., T-1\}$, then $l^G[t]$ is an optimal solution to the backward equation (17), that is,

$$f(X_{l^G[t]}[0])\mathbb{E}\left[\min\left\{C_{l^G[t]}[t], \left(A_{l^G[t]}[0] - \sum_{j=0}^{t-1} S_{l^G[t]}[j]C_{l^G[t]}[j]\right)^+\right\}\right]$$
$$+ \max_{\{\mathbf{S}[r]\}_{r=t+1}^{T-1}} \sum_{i=t+1}^{T-1} \sum_l f(X_l[0])\mathbb{E}\left[\min\left\{C_l[i], \left(A_l[0] - \sum_{j=0}^{i-1} S_l[j]C_l[j]\right)^+\right\}\right] S_l[i]$$
$$\geq f(X_m[0])\mathbb{E}\left[\min\left\{C_m[t], \left(A_m[0] - \sum_{j=0}^{t-1} S_m[j]C_m[j]\right)^+\right\}\right]$$
$$+ \max_{\{\mathbf{S}[r]\}_{r=t+1}^{T-1}} \sum_{i=t+1}^{T-1} \sum_l f(X_l[0])\mathbb{E}\left[\min\left\{C_l[i], \left(A_l[0] - \sum_{j=0}^{i-1} S_l[j]C_l[j]\right)^+\right\}\right] S_l[i], \quad (19)$$

holds for any $m \neq l^G[t]$. Recall that at most one link can be scheduled at each slot. For ease of exposition, let $\mathbf{d} \triangleq (d[t], d[t+1], ..., d[T-1])$ generically denote the sequence of feasible links chosen from time slot t to the end of the frame by any algorithm, where the element $d[i]$ denotes the link that is scheduled at slot i. Note that the elements in $\mathbf{d}$ can be any possible links. The purpose of introducing $\mathbf{d}$ is to simplify the expression of (19). Let $\mathcal{D}$ be the collection of the sequence of selected links from time slot t to the end of the frame. Let $W_\mathbf{d}$ for a given $d \in \mathcal{D}$ be defined as

$$W_\mathbf{d} \triangleq \sum_{i=t}^{T-1} f(X_{d[i]}[0])\mathbb{E}\left[\min\left\{C_{d[i]}[i], \left(A_{d[i]}[0] - \sum_{j=0}^{i-1} S_{d[i]}[j]C_{d[i]}[j]\right)^+\right\}\right]$$
$$= f(X_{d[t]}[0])\mathbb{E}\left[\min\left\{C_{d[t]}[t], \left(A_{d[t]}[0] - \sum_{j=0}^{t-1} S_{d[t]}[j]C_{d[t]}[j]\right)^+\right\}\right]$$
$$+ \sum_{i=t+1}^{T-1} \sum_l f(X_l[0])\mathbb{E}\left[\min\left\{C_l[i], \left(A_l[0] - \sum_{j=0}^{i-1} S_l[j]C_l[j]\right)^+\right\}\right] S_l[i],$$

where $S_{d[i]}[i] = 1$, and $S_l[i] = 0, \forall l \neq d[i]$, for $i = t, t+1, ..., T-1$.

Let $\mathcal{F}_l = \{\mathbf{d} \in \mathcal{D} : d[t] = l\}$. Then, (19) can be rewritten as

$$\max_{\mathbf{d} \in \mathcal{F}_{l^G[t]}} W_\mathbf{d} \geq \max_{\mathbf{d} \in \mathcal{F}_m} W_\mathbf{d}, \forall m \neq l^G[t]. \quad (20)$$

Given any $m \neq l^G[t]$, we have the following two cases:

(1) If $\mathbf{d} \in \mathcal{F}_m$ includes the element $l^G[t]$, then a permutation of $\mathbf{d}$ with the first element being $l^G[t]$ should be in $\mathcal{F}_{l^G[t]}$. Since the channel states are i.i.d. over time slots, any permutation of $\mathbf{d}$ does not change the value W_d and thus $W_\mathbf{d} \leq \max_{\mathbf{e} \in \mathcal{F}_{l^G[t]}} W_\mathbf{e}$.

(2) If $\mathbf{d} \in \mathcal{F}_m$ does not include the element $l^G[t]$, then it is easy to see that $W_\mathbf{d} \leq W_\mathbf{c}$, where $\mathbf{c} = (l^G[t], d[t+1], d[t+2], ..., d[T-1])$. Since $\mathbf{c} \in \mathcal{F}_{l^G[t]}$, we have $W_\mathbf{d} \leq \max_{\mathbf{e} \in \mathcal{F}_{l^G[t]}} W_\mathbf{e}$.

Thus, we have $\max_{\mathbf{e} \in \mathcal{F}_{l^G[t]}} W_\mathbf{e} \geq W_\mathbf{d}, \forall d \in \mathcal{F}_m$, and hence we have the desired result (20). ■

Next, we illustrate the distributed implementation of greedy solutions by using FCSMA techniques.

Idealized FCSMA Algorithm without CSI:

At each time slot $t \in \{kT, kT+1, ..., (k+1)T-1\}$ in frame k, given $(\mathbf{X}[kT], \mathbf{A}[kT])$ and $\{(\mathbf{C}[j], \mathbf{S}[j])\}_{j=kT}^{t-1}$, choose the rates

$$R_l[t] = g(X_l[kT])^{\mathbb{E}\left[\min\left\{C_l[t], \left(A_l[kT] - \sum_{j=kT}^{t-1} C_l[j]S_l[j]\right)^+\right\}\right]}, \forall l, \quad (21)$$

where $g \in \mathcal{F}$.

Note that the Idealized FCSMA Algorithm without CSI does not consider the impact of the convergence time. For the FCSMA Algorithm without CSI, the rates can be chosen as

$$R_l[t] = g(X_l[kT])^{\mathbb{E}[\min\{C_l[t], J_l[t]\}]}, \quad (22)$$

for any link l and any time $t \in \{kT, kT+1, ..., (k+1)T-1\}$, where $J_l[t]$ is the remaining data at link l at the beginning of each time slot t. Next, we will show that the Idealized FCSMA Algorithm yields the optimal performance. Simulation results indicate that both FCSMA and Idealized FCSMA Algorithm have the same performance.

Theorem 6: If $f(x) = \log g(x) \in \mathcal{B}$ and $g(0) \geq 1$, the Idealized FCSMA Algorithm without CSI for scheduling inelastic traffic is optimal over wireless fading channels, i.e., for any arrival process $\mathbf{A} \in \mathbf{\Lambda}_2(\boldsymbol{\rho}, \mathbf{C})$, it makes the underlying Markov Chain positive recurrent.

The proof is similar to that in Theorem 3 which considers the inelastic traffic with CSI over wireless fading channels. We skip it for conciseness.

B. Simulation Results

In this subsection, we perform simulations to validate the optimality of the proposed FCSMA policy without CSI for scheduling inelastic traffic with deadline constraint T slots over wireless fading channels. The simulation setup is the same as that in Section III-C. The main difference is that the fading channels change from slot to slot. The maximal satisfiable region under this setup is $\Lambda_2(\boldsymbol{\rho}, \mathbf{C}) = \{\lambda : L(1-\rho)\lambda < p\left(1-(1-\lambda)^L\right)\}$. Through numerical calculations, we can get the maximal satisfiable region: $\{\lambda : \lambda < 0.031\}$. As in Section III-C, we also compare the Idealized FCSMA Algorithm with the FCSMA Algorithm and the QCSMA algorithm with $\log\log$ function. From Figure 3, we can observe the same phenomenon as in III-C.

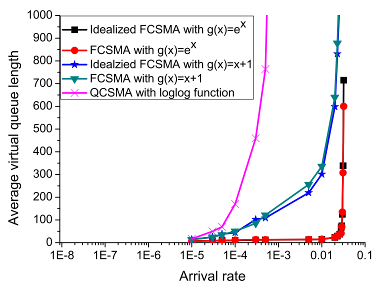


Fig. 3: The performance of the FCSMA Algorithm without CSI

V. Practical Implementation Suggestions

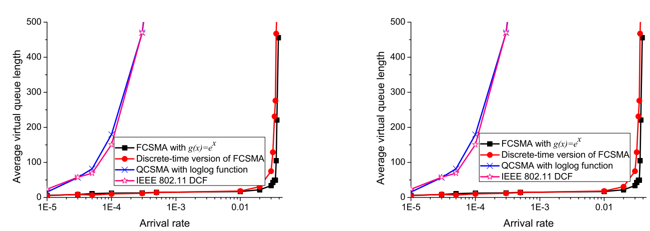
In the previous sections, we assume that the sensing is instantaneous and the backoff time is continuous, which excludes the possible collisions. These key assumptions are important in allowing us to concentrate on the challenging distributed scheduling problem in time-varying environments without considering the contention resolution procedure. Yet, in practice, the sensing time is non-zero and the backoff time is typically a multiple of mini-slots, where a mini-slot is equal to the time required to detect the data transmission from another link (e.g., in IEEE 802.11b, a mini-slot should be at least $8\mu s$). In such cases, collisions happen, which reduces the system throughput.

In this section, we explicitly consider these practical challenges and propose an easily implementable and efficient algorithm that is similar to the one in [9]. The basic idea is to quantize the continuous rate $R_l[t]$ into a set of discrete values, where each discrete value is assigned to a different contention window (CW) size. The smaller the quantized value is, the larger the corresponding CW size is. Thus, this can be easily mapped to the "service classes" in IEEE 802.11e. The suggested rate quantization procedure is as follows: (i) if $R_l[t] \geq R_{\max}$, then let $R'_l[t] = R_{\max}$. This corresponds to the first class; (ii) if $\frac{1}{2^{i-1}}R_{\max} \leq R_l[t] < \frac{1}{2^{i-2}}R_{\max}$ for some $i = 2, 3, ..., N$, then let $R'_l[t] = \frac{1}{2^{i-1}}R_{\max}$, where N is the number of classes; (iii) if $R_l[t] < R_{\min} \triangleq \frac{1}{2^{N-1}}R_{\max}$, then do not start transmissions. Thus, the probability of links accessing the channel in class i is roughly twice as large as that in class $i+1$, which implies that the CW size of class $i+1$ should be roughly twice that of class i.

Discrete-time version of the FCSMA Algorithm:

At the beginning of each slot t, each link l generates a uniformly distributed random variable r_l over $\{0, 1, ..., \text{CW}[t]-1\}$, where CW$[t]$ is chosen according to the quantized value of rate $R_l[t]$ as described above. Each link l keeps sensing the channel for r_l mini-slots. If the channel is busy in any one of the first r_l mini-slots, then link l suspends its transmission; otherwise, link l starts its transmission[3] from the r_l^{th} mini-slot to the end of this slot. If two or more links have the same backoff time, then a collision happens in the current slot. The whole process restarts in the next slot.

We assume that the coherence time for scheduling inelastic traffic with and without CSI are $500ms$ and $100ms$, respectively. Since a mini-slot is typically $10\mu s$, without loss of generality, we assume that a time slot contains 10000 mini-slots in both cases. The simulation setups for scheduling inelastic traffic with and without CSI are the same as that in Section III-C and Section IV-B, respectively. In the simulations, we let $R_{\max} = e^5$, $N = 6$, and $\text{CW}_i = 32 \times 2^{i-1}, i = 1, 2, ..., N$, where CW_i is the CW of class i. We also compare the discrete-time version of the FCSMA Algorithm with IEEE 802.11 Distributed Coordination Function (DCF). In IEEE 802.11 DCF, the contention window (CW) size depends on whether the transmission is successful or not, rather than the current system state information. In particular, the CW for all links are initialized to 32; if the transmission of link l is unsuccessful, then its CW is doubled until it reaches to the maximum value of 1024; otherwise, its CW drops to the initial value.



(a) Scheduling inelastic traffic with CSI (b) Scheduling inelastic traffic without CSI

Fig. 4: Performance comparison between FCSMA algorithm and its discrete-time version

From Figure 4a and 4b, we can observe that the performance of the discrete-time version of the FCSMA Algorithm remains close to that of the FCSMA Algorithm, and continue to perform much better than the QCSMA algorithm with $\log\log$ function

[3] If the number of links is large, each link uses short packets, such as Request-To-Send (RTS) and Clear-To-Send (CTS) in IEEE 802.11b, to contend for the wireless channel, which will significantly reduce the cost of a collision.

and IEEE 802.11 DCF in both scheduling inelastic traffic with and without CSI. However, we note that if the coherent time is comparable with the maximum CW size, then, the discrete-time version of the FCSMA Algorithm can perform poorly, since a non-negligible amount of resources is consumed by the backoff process instead of the data transmission.

VI. Conclusions

In this paper, we first proposed a Fast-CSMA (FCSMA) Algorithm that quickly reaches the favorable state in fully connected network topologies. Due to the fast convergence time, the FCSMA Algorithm exhibits significant advantages over existing CSMA algorithms for time-varying applications, which are important and popular in wireless networks. Then, we apply the FCSMA Algorithm to design optimal policies for scheduling inelastic traffic with/without CSI over wireless fading channels. In the future, we will try to explore distributed scheduling algorithms for time-varying environments in general wireless network topologies.

Appendix A
Proof for Theorem 3

Consider the Lyapunov function $V(\mathbf{X}) \triangleq \sum_{l=1}^{L} h(X_l)$, where $h'(x) = f(x)$. Then, by using a similar argument to the proof of Lemma 1 in [28] (also see [29]), it is not hard to show that if for any process $\mathbf{A} \in \Lambda_1(\boldsymbol{\rho}, \mathbf{C})$, there exists $\gamma > 0$ and $H \geq 0$ such that

$$\begin{aligned}\Delta V(\mathbf{X}) \triangleq& \sum_{l=1}^{L} \mathbb{E}\left[f(X_l)(D_l[kT] - I_l[kT]) | \mathbf{X}[kT] = \mathbf{X}\right] \\ \leq& -\gamma \sum_{l=1}^{L} f(X_l) + H. \end{aligned} \tag{23}$$

By the telescoping technique, we have

$$\limsup_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{L} \mathbb{E}[f(X_l[kT])] \leq \frac{H}{\gamma} < \infty,$$

which implies the stability-in-the-mean and thus the Markov Chain is positive recurrent [30]. Next, we will show inequality (23) to complete the proof. By substituting the expression of $D_l[kT]$ (see the discussion before (4)) into $\Delta V(\mathbf{X})$, we have

$$\begin{aligned}\Delta V(\mathbf{X}) =& \underbrace{\sum_{l=1}^{L} \mathbb{E}\left[f(X_l)(A_l[kT] - I_l[kT]) | \mathbf{X}[kT] = \mathbf{X}\right]}_{\triangleq \Delta V_1(\mathbf{X})} \\ & \underbrace{- \mathbb{E}\left[\sum_{l=1}^{L} f(X_l) \min\left\{\sum_{t=kT}^{(k+1)T-1} C_l[kT] S_l^F[t], A_l[kT]\right\} \middle| \mathbf{X}[kT]\right]}_{\triangleq \Delta V_2(\mathbf{X})},\end{aligned}$$

where $\mathbf{S}^F[t] = (S_l^F[t])_{l=1}^L$ denotes the schedule chosen by FCSMA algorithm at time t. Let

$$W_l[t] = f(X_l[kT]) \min\left\{C_l[kT], \left(A_l[kT] - C_l[kT] \sum_{j=kT}^{t-1} S_l[j]\right)^+\right\},$$

for any $t = kT, kT+1, ..., (k+1)T-1$, where $\mathbf{S}[j] = (S_l[j])_{l=1}^L$ is a feasible schedule at time slot j. Let $W^G[t]$ be the weight of link picked by the Greedy Algorithm with CSI at time slot t. Recall that $W^G[t] = \max_l W_l[t]$. Next, we will derive an upper bound for $\Delta V_1(\mathbf{X})$ by using Lemma 2 and give a lower bound for $\Delta V_2(\mathbf{X})$.

First, let's focus on ΔV_1. By Lemma 2, there exist non-negative numbers $\alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1})$ satisfying (5) and for a $\delta > 0$ small enough, we have

$$\begin{aligned}\lambda_l(1-\rho_l) \leq& \sum_{\mathbf{a}} P_{\mathbf{A}}(\mathbf{a}) \sum_{\mathbf{c}} P_{\mathbf{C}}(\mathbf{c}) \sum_{\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1} \in \mathcal{S}} \\ & \alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1}) \min\left\{\sum_{j=0}^{T-1} c_l s_l^j, a_l\right\} - \delta. \end{aligned} \tag{24}$$

By using (24), we have

$$\begin{aligned}\Delta V_1 =& \sum_{l=1}^{L} f(X_l)\lambda_l(1-\rho_l) \\ \leq& \sum_{\mathbf{a}} P_{\mathbf{A}}(\mathbf{a}) \sum_{\mathbf{c}} P_{\mathbf{C}}(\mathbf{c}) \sum_{\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1} \in \mathcal{S}} \alpha(\mathbf{a}, \mathbf{c}; \mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{T-1}) \\ & \sum_{l=1}^{L} f(X_l) \min\left\{\sum_{j=0}^{T-1} c_l s_l^j, a_l\right\} - \delta \sum_{l=1}^{L} f(X_l) \\ \leq& \sum_{\mathbf{a}} P_{\mathbf{A}}(\mathbf{a}) \sum_{\mathbf{c}} P_{\mathbf{C}}(\mathbf{c}) \sum_{t=kT}^{(k+1)T-1} W^G[t] - \delta \sum_{l=1}^{L} f(X_l), \end{aligned} \tag{25}$$

where the last step follows from Theorem 2 that the Greedy Algorithm with CSI maximizes $\sum_{l=1}^{L} f(X_l) \min\left\{\sum_{j=0}^{T-1} c_l s_l^j, a_l\right\}$ for any feasible schedules $\mathbf{s}^0$, $\mathbf{s}^1$,...,$\mathbf{s}^{T-1}$, given virtual queue lengths, channel state information and arrivals, and uses (5).

Thus, we have

$$\begin{aligned}\Delta V_1 \leq& \mathbb{E}\left[\sum_{t=kT}^{(k+1)T-1} W^G[t] \middle| \mathbf{X}[kT] = \mathbf{X}\right] - \delta \sum_{l=1}^{L} f(X_l) \\ =& \mathbb{E}\left[\mathbb{E}\left[\sum_{t=kT}^{(k+1)T-1} W^G[t] \middle| \mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT]\right] \middle| \mathbf{X}[kT]\right] \\ & - \delta \sum_{l=1}^{L} f(X_l). \end{aligned} \tag{26}$$

Note that $W^G[t]$ is non-increasing in t within each frame, since the number of remaining packets cannot increase as t increases.

Pick any $\overline{W} > 0$ and let

$\mathcal{F}_0 \triangleq \{W^G[kT] \leq \overline{W}, W^G[kT+1] \leq \overline{W}, ..., W^G[(k+1)T-1] \leq \overline{W}\};$
$\mathcal{F}_j \triangleq \{W^G[kT+j-1] > \overline{W}, W^G[kT+j] \leq \overline{W}\}, \forall j = 1, ..., T-1;$
$\mathcal{F}_T \triangleq \{W^G[(k+1)T-1] > \overline{W}\},$

where $\mathcal{F}_j$ corresponds to the event where the weight chosen by Greedy Algorithm is greater than $\overline{W}$ in the first j slots in frame k. Thus, $(\mathcal{F}_j)_{j=0}^T$ forms a partition of a set $\{W^G[kT], W^G[kT+1], ..., W^G[(k+1)T-1]\}$. Then, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=kT}^{(k+1)T-1} W^G[t] \middle| \mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT]\right] \\
=&\mathbb{E}\left[\sum_{j=0}^{T} \sum_{t=kT}^{(k+1)T-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT]\right] \\
\leq&\mathbb{E}\left[\sum_{j=0}^{T} \left(\sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} + (T-j)\overline{W}\right) \middle| \mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT]\right] \\
=&\mathbb{E}\left[\sum_{j=1}^{T} \sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT], \mathbf{A}[kT], \mathbf{C}[kT]\right] + \frac{T(T+1)\overline{W}}{2}.
\end{aligned}$$

Thus, ΔV_1 becomes

$$\begin{aligned}
\Delta V_1 \leq&\mathbb{E}\left[\sum_{j=1}^{T} \sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT] = \mathbf{X}\right] + \frac{T(T+1)\overline{W}}{2} \\
&- \delta \sum_{l=1}^{L} f(X_l). \qquad (27)
\end{aligned}$$

Second, let's consider ΔV_2. Let

$$W_l^F[t] = f(X_l[kT]) \min\left\{C_l[kT], \left(A_l[kT] - C_l[kT]\sum_{j=kT}^{t-1} S_l^F[j]\right)^+\right\}.$$

Then, by using Lemma 3 and switching the summations over l and t, we have

$$\Delta V_2 = \mathbb{E}\left[\sum_{t=kT}^{(k+1)T-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t] \middle| \mathbf{X}[kT] = \mathbf{X}\right]. \qquad (28)$$

Let $\epsilon > 0$ and $\zeta > 0$. For each event $\mathcal{F}_j, \forall j = 1, 2, ..., T$, we have $W^G[kT] > \overline{W}, ..., W^G[kT+j-1] > \overline{W}$. By using Lemma 1, we obtain that for any $\zeta' > 0$, choose $\overline{W}$ such that

$$\Pr\left\{\sum_{l=1}^{L} W_l^F[t]S_l^F[t] \geq (1-\epsilon)W^G[t] \middle| \mathcal{F}_j\right\} \geq 1 - \zeta', \qquad (29)$$

holds for any $t = kT, kT+1, ..., kT+j-1$. Hence, we have

$$\begin{aligned}
&\Pr\left\{\sum_{t=kT}^{kT+j-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t] \geq (1-\epsilon)\sum_{t=kT}^{kT+j-1} W^G[t] \middle| \mathcal{F}_j\right\} \\
&\geq \Pr\left\{\sum_{l=1}^{L} W_l^F[t]S_l^F[t] \geq (1-\epsilon)W^G[t], \forall t = kT, ..., kT+j-1 \middle| \mathcal{F}_j\right\} \\
&\geq 1 - j\zeta' \geq 1 - T\zeta', \qquad (30)
\end{aligned}$$

where we use the fact that given any two events $\mathcal{E}_1$ and $\mathcal{E}_2$ such that $\Pr\{\mathcal{E}_1\} \geq 1 - \epsilon_1$ and $\Pr\{\mathcal{E}_2\} \geq 1 - \epsilon_2$, we have $\Pr\{\mathcal{E}_1 \bigcap \mathcal{E}_2\} \geq 1 - \epsilon_1 - \epsilon_2$. By picking ζ' small enough such that $1 - T\zeta' \geq 1 - \zeta$, we have

$$\Pr\left\{\sum_{t=kT}^{kT+j-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t] \geq (1-\epsilon)\sum_{t=kT}^{kT+j-1} W^G[t] \middle| \mathcal{F}_j\right\} \geq 1 - \zeta,$$

for $j = 1, ..., T$, which implies that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=kT}^{kT+j-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT] = \mathbf{X}\right] \\
&= \Pr\{\mathcal{F}_j\}\mathbb{E}\left[\sum_{t=kT}^{kT+j-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t] \middle| \mathbf{X}[kT] = \mathbf{X}, \mathcal{F}_j\right] \\
&\geq \Pr\{\mathcal{F}_j\}(1-\epsilon)(1-\zeta)\mathbb{E}\left[\sum_{t=kT}^{kT+j-1} W^G[t] \middle| \mathbf{X}[kT] = \mathbf{X}, \mathcal{F}_j\right] \\
&=(1-\epsilon)(1-\zeta)\mathbb{E}\left[\sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT] = \mathbf{X}\right], \qquad (31)
\end{aligned}$$

for $j = 1, ..., T$. Thus, we have

$$\begin{aligned}
\Delta V_2 &= \mathbb{E}\left[\sum_{j=0}^{T} \sum_{t=kT}^{(k+1)T-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT] = \mathbf{X}\right] \\
&\geq \mathbb{E}\left[\sum_{j=1}^{T} \sum_{t=kT}^{kT+j-1} \sum_{l=1}^{L} W_l^F[t]S_l^F[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT] = \mathbf{X}\right] \\
&\geq (1-\epsilon)(1-\zeta)\mathbb{E}\left[\sum_{j=1}^{T} \sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT]\right] \qquad (32)
\end{aligned}$$

Thus, by using (27) and (32), ΔV becomes

$$\begin{aligned}
\Delta V \leq&(\epsilon + \zeta - \epsilon\zeta)\mathbb{E}\left[\sum_{j=1}^{T} \sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT]\right] \\
&- \delta \sum_{l=1}^{L} f(X_l) + \frac{T(T+1)\overline{W}}{2}. \qquad (33)
\end{aligned}$$

Since

$$\begin{aligned}
&\mathbb{E}\left[\sum_{j=1}^{T} \sum_{t=kT}^{kT+j-1} W^G[t]\mathbb{1}_{\mathcal{F}_j} \middle| \mathbf{X}[kT]\right] \\
\leq&\mathbb{E}\left[\sum_{t=kT}^{(k+1)T-1} W^G[t] \middle| \mathbf{X}[kT]\right] \leq A_{\max}T\sum_{l=1}^{L} f(X_l), \qquad (34)
\end{aligned}$$

we have

$$\begin{aligned}
&\Delta V \leq (\epsilon + \zeta - \epsilon\zeta)A_{\max}T\sum_{l=1}^{L} f(X_l) - \delta\sum_{l=1}^{L} f(X_l) + \frac{T(T+1)\overline{W}}{2} \\
&= -\gamma\sum_{l=1}^{L} f(X_l) + H, \qquad (35)
\end{aligned}$$

where $H = \frac{T(T+1)\overline{W}}{2}$ and $\gamma = \delta - A_{\max}(\epsilon + \zeta - \epsilon\zeta)T$. We can choose β, ϵ, ζ small enough such that $\gamma > 0$.

References

[1] B. Li and A. Eryilmaz. A fast-csma algorithm for deadline-constrained scheduling over wireless fading channels. In *Proc. International Workshop on Resource Allocation and Cooperation in Wireless Networks (RAWNET/WNC3)*, Princeton, New Jersey, May 2011.

[2] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policiesfor maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 36(12):1936–1948, 1992.

[3] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queueswith randomly varying connectivity. *IEEE Transactions on Information Theory*, 39(2):466–478, 1993.

[4] A. Eryilmaz, R. Srikant, and J. Perkins. Stable scheduling policies for fading wireless channels. *IEEE/ACM Transactions on Networking*, 13(2):411–424, 2005.

[5] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, California, April 2003.

[6] S. Shakkottai and A. Stolyar. Scheduling for multiple flows sharing atime-varying channel: The exponential rule. *Translations of the AMS, Series 2,* Avolume in memory of F. Karpelevich, 207:185–202, 2002.

[7] A. Eryilmaz, A. Ozdaglar, D. Shah, and E. Modiano. Distributed cross-layer algorithms for the optimal control of multihop wireless networks. *IEEE/ACM Transactions on Networking*, 18(2):638–651, 2010.

[8] B. Li and A. Eryilmaz. On the limitation of randomization for queue-length-based scheduling in wireless networks. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Shanghai, China, April 2011.

[9] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, June 2010.

[10] 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 *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, mini-Conference, San Diego, CA, March 2010.

[11] J. Ghaderi and R. Srikant. On the design of efficient csma algorithms for wireless networks. In *Proc. IEEE International Conference on Decision and Control (CDC)*, Atlanta, GA, December 2010.

[12] S. Rajagopalan, D. Shah, and J. Shin. Network adiabatic theorem: an efficient randomized protocol for contention resolution. In *Proc. IEEE International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Seattle, WA, June 2009.

[13] D. Shah, D. Tse, and J. Tsitsiklis. Hardness of low delay network scheduling. *IEEE Transactions on Information Theory*, 57(12):7810–7817, December 2011.

[14] M. Lotfinezhad and P. Marbach. Throughput-optimal random access with order-optimal delay. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Shanghai, China, April 2011.

[15] B. Li and A. Eryilmaz. Optimal distributed scheduling under time-varying conditions: A fast-csma algorithm with applications. Technical Report, 2012. Available online at http://www.ece.osu.edu/∼eryilmaz/fcsma_report.pdf.

[16] I. Hou, V. Borkar, and P. R. Kumar. A theory of QoS for wireless. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 2009.

[17] I. Hou and P. R. Kumar. Scheduling heterogeneous real-time traffic over fading wireless channels. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, San Diego, CA, March 2010.

[18] J. Jaramillo, R. Srikant, and L. Ying. Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints. *IEEE Journal on Selected Areas in Communications*, 29(5):979–987, 2011.

[19] H. Gangammanavar and A. Eryilmaz. Dynamic coding and rate-control for serving deadline-constrained traffic over fading channels. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, Austin, TX, June 2010.

[20] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Miami, Florida, March 2005.

[21] P. Marbach, A. Eryilmaz, and A. Ozdaglar. Achievable rate region of csma schedulers in wirelessnetworks with primary interference constraints. In *Proc. IEEE Conference on Decision and Control (CDC)*, New Orleans, Louisiana, December 2007.

[22] S. Rajagopalan and D. Shah. Distributed algorithm and reversible network. In *Proc. IEEE Conference on Information Sciences and Systems (CISS)*, Princeton, New Jersey, March 2008.

[23] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networksand input queued switches. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, California, April 1998.

[24] M. Neely. *Stochastic Network Optimization with Aapplication to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[25] J. Dai. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability*, 5:49–77, 1995.

[26] L. Tassiulas. Scheduling and performance limits of networks with constantly varyingtopology. *IEEE Transactions on Information Theory*, 43:1067–1073, May 1997.

[27] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific; 3rd edition, 2007.

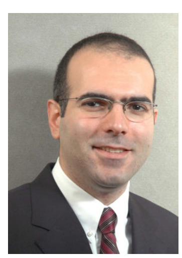
[28] B. Li and A. Eryilmaz. On the boundaries of randomization for throughput-optimal scheduling in wireless networks. In *Proc. Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, Illinois, September 2010.

[29] B. Li and A. Eryilmaz. Exploring the throughput boundaries of randomized schedulers in wireless networks. *IEEE/ACM Transactions on Networking*, 20(4):1112–1124, 2012.

[30] S. Meyn and R. Tweedie. Criteria for stability of markovian processes i: Discrete time chains. *Advances in Applied Probability*, 24:542–574, 1992.

Bin Li received his B.S. degree in Electronic Engineering and M.S. degree in Communication Engineering, both from Xiamen University, China, in 2005 and 2008, respectively. Between 2008 and 2009, he was with the University of Texas at Arlington. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at The Ohio State University.

His research interests include wireless communication and networks, optimal network control, distributed algorithms, and optimization theory.



Atilla Eryilmaz (S ’00-M ’06) received his B.S. degree in Electrical and Electronics Engineering from Boğaziçi University, Istanbul, in 1999, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2001 and 2005, respectively. Between 2005 and 2007, he worked as a Postdoctoral Associate at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. He is currently an Associate Professor of Electrical and Computer Engineering at The Ohio State University.

Dr. Eryilmaz’s research interests include design and analysis for communication networks, optimal control of stochastic networks, optimization theory, distributed algorithms, pricing in networked systems, and information theory. He received the NSF-CAREER and the Lumley Research Awards in 2010.