This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2343613, IEEE Transactions on Mobile Computing

Throughput-Optimal Queue Length Based CSMA/CA Algorithm for Cognitive Radio Networks

Shuang Li, *Student Member, IEEE,* Eylem Ekici, *Senior Member, IEEE,* and Ness Shroff, *Fellow, IEEE*

***Abstract*—Cognitive Radio Networks allow unlicensed users to access licensed spectrum opportunistically without disrupting primary user (PU) communication. Developing a distributed implementation that can fully utilize the spectrum opportunities for secondary users (SUs) has so far remained elusive. Although throughput optimal algorithms based on the well-known Maximal Weight Scheduling (MWS) algorithm exist for cognitive radio networks, they require central processing of network-wide SU information. In this paper, a new distributed algorithm is introduced that asymptotically achieves the capacity region of the cognitive radio systems. The proposed algorithm achieves the full SU capacity region while adapting to the channel availability dynamics caused by unknown Primary User (PU) activity. Extensive simulation results are provided to illustrate the efficacy of the algorithm.**

***Index Terms*—Cognitive Radio Networks, CSMA, throughput optimal, distributed.**

I. Introduction

Cognitive radio networks (CRNs) allow unlicensed or secondary users to opportunistically access licensed spectra to improve the overall utilization of wireless resources and address the problem of spectrum shortage. The design of CRNs poses new challenges that are not present in traditional wireless networks [6]. SUs are allowed spectrum access only when they do not cause unacceptable levels of interference to licensed or primary users (PUs). DARPA's 'Next Generation' (XG) program [15] mandates that cognitive radios sense signals to prevent interference to existing military and civilian radio systems. On the other hand, IEEE regulates unlicensed access to the TV spectrum without interference, as described in the IEEE 802.22 standard [7]. These and other interference avoidance mechanisms profoundly impact the design of algorithms and protocols for cognitive radio networks.

The complete utilization of the so-called spectrum opportunities while avoiding interference on PUs is the goal of algorithm and protocol design for cognitive radio networks. In traditional wireless networks, Maximum Weight Scheduling (MWS) algorithm [16] and its variants achieve the full capacity region of the network, where a scheduling policy is said to achieve the full capacity region (or be throughput optimal) [9] if it stabilizes the system for any arrival rate vector the system can be stabilized for by some scheduling policy. However, these algorithms require the knowledge of the entire network state and centralized processing to compute conflict free schedules. Similar algorithms have also been proposed for cognitive radio networks: In [17], opportunistic scheduling policies are developed for multichannel single-hop CRNs subject to maximum collision rate constraints with PUs. In [18], scheduling algorithms are investigated in multi-channel multi-hop CRN overlayed with a PU network. The optimal throughput can be provably and asymptotically achieved in adaptive-routing scenarios. Both works require solving an NP-hard problem centrally.

These centralized throughput optimal algorithms suffer from two main shortcomings. The first is the high computational complexity, and the second one is the cost associated with the collection of network state information at a central location. The first problem has been countered in the literature through lower complexity suboptimal algorithms. Maximal Scheduling is such an algorithm that re-

Shuang Li is with the Department of Computer Science and Engineering, Ohio State University, Columbus, OH, 43210 USA e-mail: li.908@osu.edu. This work has been funded by ARO MURI award W911NF-08-1-0238.

Eylem Ekici is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH, 43210 USA e-mail: ekici@ece.osu.edu.

Ness Shroff is with the Department of Electrical and Computer Engineering and Computer Science and Engineering, Ohio State University, Columbus, OH, 43210 USA e-mail: shroff@ece.osu.edu.

duces the time complexity of MWS at the expense of achieving a small fraction of the capacity region [1]. Another algorithm, Greedy Maximal Scheduling (GMS), always selects a link with the longest queue that does not cause interference to links already chosen. GMS is shown to be throughput optimal if the network topology satisfies the local-pooling condition [2]. Nevertheless, only a fraction of the capacity region can be achieved for general topologies [9]. Furthermore, the distributed implementation of GMS entails signaling overhead which is not scalable [11].

Recently, a new class of distributed algorithms has been proposed to achieve throughput optimality while circumventing these problems. These new algorithms are random access algorithms based on the notion of channel sensing. These algorithms use queue lengths to determine channel access probabilities, achieving the full capacity region in ad hoc wireless networks in a distributed manner. In [8], an adaptive throughput optimal CSMA scheduling algorithm is proposed for a general interference model in continuous time. It uses transmission aggressiveness, which is a function of the queue length. Implementation considerations in 802.11 networks are discussed considering packet collisions. In [13], Q-CSMA, a discrete-time distributed randomized algorithm based on Glauber dynamics is proposed. In both [8] and [13], the queue length based CSMA algorithms achieve the full capacity region in a single-channel ad hoc wireless network.

The queue length-based CSMA algorithms of [8] [13] assume that the channel is always available, a condition not satisfied in cognitive radio networks. With the randomness of the channel availability, state transitions are forced by the channel state going from ON to OFF. In this paper, we develop CA-CSMA (Channel-aware CSMA/CA), a new distributed throughput optimal scheduling algorithm for CRNs. To this end, we introduce a new system state representation that includes channel state information, and design our algorithms to achieve throughput optimality without causing interference with PUs. In this work, we focus on the algorithm design for single-channel cognitive radio networks. Our analysis can be immediately extended to a system with orthogonal channels, where scheduling is performed per channel. However, extensions to non-orthogonal channels are beyond the scope of this paper.

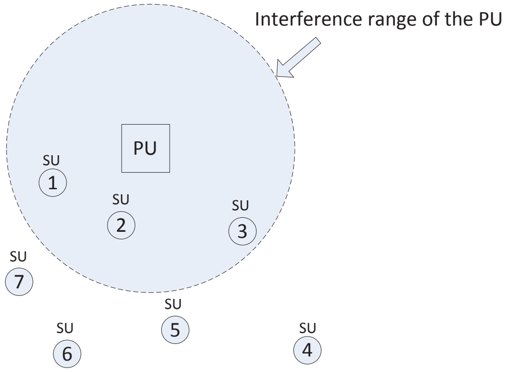

Fig. 1. A SU network composed of 7 SUs overlayed with a PU network.

The paper is organized as follows: In Section II, the system model is introduced. In Section III, we discuss some technical challenges that require a new state space representation and then describe the appropriate representation on which CA-CSMA is built. Numerical performance evaluations are presented in Section IV. The paper is concluded in Section V.

II. System Model

We consider a cognitive radio network consisting of N SUs coexisting with a PU network (Figure 1), where the PU network is represented as a single source of emission and the SUs communicate with their neighbors directly. The PU network has a single designated channel to transmit on. We adopt the exclusive communication approach to interference avoidance, i.e., an SU can transmit only when the PU network does not use the channel. The set S denotes all SUs that are outside the interference range of the PU network while $\bar{S}$ is the set of all other SUs. i.e., SUs 1-3 are in $\bar{S}$ and SUs 4-7 are in S. SUs in S have access to the channel at any time since its transmission does not interfere with the PU network. SUs in $\bar{S}$ can sense the channel and keep silent when the PU network is active.

The neighboring SUs of a given SU i ($i = 1, \cdots, N$) is denoted by $C(i)$ such that i and any of its neighbors $j \in C(i)$ cannot transmit at the same time. Note that both the k-hop ($k \geq 1$) [9] and distance-based [5] interference models are included as special cases. Symmetry is assumed in the conflict set, i.e., if $i \in C(j)$, then $j \in C(i)$.

We consider a time-slotted system with unit capacity links. A feasible schedule includes SUs that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2343613, IEEE Transactions on Mobile Computing

can be active at the same time subject to the conflict set constraints. Let M be the set of all feasible schedules. A schedule is represented by a set x ($x \in M$) and we denote the vector of schedule by $\vec{x'} \in \{0,1\}^N$ such that $x'_i = 1$ if SU i is in the schedule x. So $i \in x$ indicates that $x'_i = 1$. The capacity region of the SUs, in the absence of PU activity is defined as $\Lambda = \{\vec{\lambda}|\vec{\lambda} \succeq \vec{0}$ and $\exists \vec{\mu} \in Co(M), \vec{\lambda} \prec \vec{\mu}\}$, where $Co(\cdot)$ is the convex hull and $\succeq$, $\prec$ are component-wise "greater than or equal to" and "smaller than" operators, respectively. The actual capacity region of SUs is the interior of Λ subject to the PU activity.

In this work, single-hop flows for both the primary and the secondary systems are considered. We assume that the PU has i.i.d. Bernoulli traffic in each time slot, where the PU is idle (channel is available) with probability p. $B(t)$ is defined as the channel state at time t, where $B(t) = 0$ if the channel is available (PU is idle) and $B(t) = 1$ if the channel is unavailable (PU is active).

III. The Distributed Scheduling Algorithm for CRNs

A. Q-CSMA Overview

As mentioned in the introduction, throughput optimal CSMA-based distributed scheduling algorithms such as Q-CSMA [13] have been proposed in the recent past. It is tempting to apply such algorithms to achieve throughput optimality in a distributed manner although it does not consider cognitive radio environments. In the following, we first present an overview of the Q-CSMA algorithm. We then demonstrate why Q-CSMA cannot be directly applied to CRNs, which motivates our main contributions in this work.

1) Introduction to Q-CSMA: In [13], a discrete-time distributed randomized algorithm is proposed to achieve the full capacity region in a single-channel network. The algorithm of [13] is based on a generalization of Glauber dynamics in statistical physics. In Glauber dynamics, only one link has a state update within a time slot. In scheduling, a state update can be interpreted as a transition of a link from "transmitting" to "idle" or from "idle" to "transmitting". The incremental state update in every time slot leads to a scheduling policy sufficiently close to MWS, which guarantees the throughput optimality. In [13], multiple links are allowed to update their states in a single time slot. This minor change, which results in improved delay performance, does not affect throughput optimality.

A more detailed description of the Q-CSMA algorithm is as follows: Each time slot t is divided into a control slot and a data slot, where the control slot is much smaller than the data slot. In the control slot, a collision-free transmission schedule is generated and used for data transmission in the data slot. Let $m(t)$ be a set of SUs that do not conflict with each other and selected randomly in the control slot (the scheme will be presented in Section III-B). M_0 denotes the set of all $m(t)$ which is all possible schedules. So M_0 includes all feasible schedules that could be generated by the randomized algorithm. Note that $M_0 \subseteq M$, the set of all feasible schedules. The network randomly selects a feasible schedule $m(t)$, which is called the decision schedule in [13]. $m(t)$ can be regarded as a candidate schedule. Note that $m(t)$ and $m(t-1)$ are independent for all $t > 0$ because $m(t)$ is chosen independently in the subsequent control slot [13]. Each link within $m(t)$ will be checked to decide whether it will be included in the transmission schedule $x(t)$. Link $i \in m(t)$ may be included in $x(t)$ if $\forall j \in C(i)$, $j \notin x(t-1)$; otherwise, Link i is not included in $x(t)$. Furthermore, link $k \notin m(t)$ is included in $x(t)$ if $k \in x(t-1)$. The detailed algorithm is as follows: links in $m(t)$ that had no neighbors active in the previous data slot are allowed to update their states with a certain probability which is a function of their queue lengths; those outside the decision schedule $m(t)$ maintain their states. By explicitly taking into account collisions in the control slot, the algorithm generates collision-free transmission schedules $x(t)$ for the data slot. More importantly, the Discrete Time Markov Chain (DTMC) with the transmission schedule chosen as the state is shown to be time-reversible and has product-form stationary distribution, which are used to prove throughput optimality of this algorithm.

The operation of the Q-CSMA algorithm is illustrated in Figure 2 for a simple two-link topology, where the two links interfere with each other. State $(0, 0)$ represents that neither link transmits and state $(0, 1)$ indicates that only link 2 transmits. α_i is the probability that link i ($i = 1, 2$) is chosen in the decision schedule $m(t)$ and p_i is the probability that link i is activated given it is in $m(t)$ and no neighbors were active in the previous

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2343613, IEEE Transactions on Mobile Computing

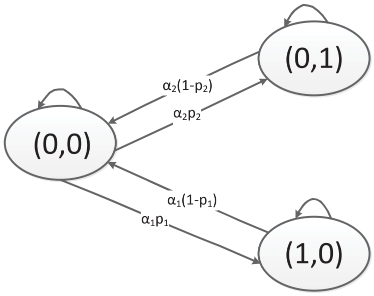

Fig. 2. DTMC with the vector of transmission schedule $\vec{x'}(t)$ of two links as the state. Two links interfere with each other.

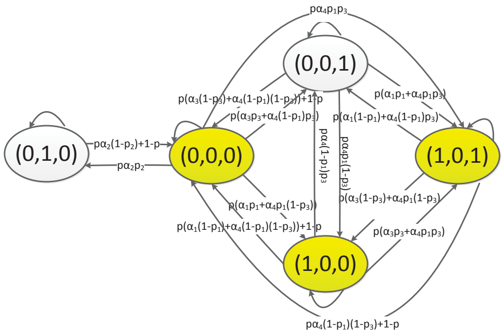
$p\alpha_4 p_1 p_3$
(0,0,1)
$p(\alpha_3(1-p_3)+\alpha_4(1-p_1)(1-p_3))+1-p$
$p(\alpha_3 p_3+\alpha_4(1-p_1)p_3)$
$p(\alpha_1 p_1+\alpha_4 p_1 p_3)$
$p(\alpha_1(1-p_1)+\alpha_4(1-p_1)p_3)$
(0,1,0)
$p\alpha_2(1-p_2)+1-p$
$p\alpha_2 p_2$
(0,0,0)
$p\alpha_4(1-p_1)p_3$
$p\alpha_4 p_1(1-p_3)$
(1,0,1)
$p(\alpha_3(1-p_3)+\alpha_4 p_1(1-p_3))$
$p(\alpha_1 p_1+\alpha_4 p_1(1-p_3))$
$p(\alpha_1(1-p_1)+\alpha_4(1-p_1)(1-p_3))+1-p$
$p(\alpha_3 p_3+\alpha_4 p_1 p_3)$
(1,0,0)
$p\alpha_4(1-p_1)(1-p_3)+1-p$

Fig. 3. DTMC with the vector of transmission schedule $\vec{x'}(t)$ of three SU links as the state in CRNs. 1 and 2 interfere with each other. 2 and 3 interfere with each other. 1 and 3 can transmit simultaneously.

data slot. The DTMC is time reversible and has the following product-form stationary distribution: $\pi(0,0) = \frac{1}{Z}$, $\pi(0,1) = \frac{1}{Z}\frac{p_2}{1-p_2}$, $\pi(1,0) = \frac{1}{Z}\frac{p_1}{1-p_1}$ where $Z = 1 + \frac{p_1}{1-p_1} + \frac{p_2}{1-p_2}$ and $\pi(a,b)$ is the stationary distribution for state (a,b) (a, $b = 0$, 1), which are in accordance with Proposition 1 of [13].

2) Drawback of Q-CSMA in CRNs: We next illustrate why $x(t)$ is a poor choice for representing the state in CRNs, and leads to a non-reversible DTMC. For clarity, we use $\vec{x'}(t)$, the vector of transmission schedule as defined in Section II, as the state. Consider a simple example in Figure 3 for three interfering SU links, where 1 interferes with 2, 2 interferes with 3 and 1 does not interfere with 3. All SU links are within the interference range of the PU. Similar to the earlier example, 1 (0) indicates that an SU link is (not) transmitting. For instance, state $(1,0,1)$ means that SU links 1 and 3 are transmitting and SU link 2 is not transmitting. Note that $(0,0,0)$ includes two cases: 1) The channel is available and no SU link is transmitting; 2) The channel is unavailable. Let α_i be the probability that SU link i is chosen in the decision schedule $m(t)$ ($i = 1$, 2, 3), α_4 the probability that both 1 and 3 are chosen in $m(t)$, and p_i the probability that SU link i is activated ($i = 1$, 2, 3) given it is in $m(t)$ and no neighbors were active in the previous data slot. Now we examine the transitions from $(0,0,0)$, $(1,0,0)$ to $(1,0,1)$ clockwise and counter-clockwise. These two probabilities are not equal so the DTMC is not time reversible by Kolmogorov's criterion [10].

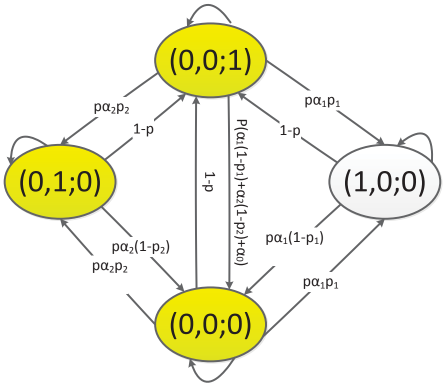

Fig. 4. DTMC with both the vector of transmission schedule $\vec{x'}(t)$ of two SUs and channel state as the state. Two SUs are in the conflict set of each other.

To treat the "available" and "unavailable" channel separately, we incorporate the channel state into the state space design. The new DTMC is illustrated in Figure 4 where we consider two SUs interfering with each other and both are in the interference range of the PU. The new system state is defined as $X' = (\vec{x'}; B)$ where $\vec{x'}$ is the vector of transmission schedule and B is the channel state defined in Section II (0 represents that the channel is idle; 1 represents that the channel is busy). For instance, $(0,0;1)$ indicates that the channel is unavailable and neither SU is transmitting; $(0,1;0)$ means the channel is available and SU 2 is transmitting. Note that α_1 and α_2 have the same definitions as before; α_0 is the probability that neither SU is selected in the decision schedule ($\alpha_0+\alpha_1+\alpha_2 = 1$). We now check to see whether the DTMC is time reversible by examining the transitions from $(0,0;1)$, $(0,0;0)$ to $(0,1;0)$ clockwise and counter-clockwise. The product of clockwise transition probabilities is $p^2(1-p)p_2\alpha_2(\alpha_1(1-p_1)+\alpha_2(1-p_2)+\alpha_0)$ and the product of counter-clockwise transition probabilities is $p^2(1-p)\alpha_2{}^2 p_2(1-p_2)$. These two are not equal so the DTMC is not time reversible by Kolmogorov's criterion [10].

3) The Necessity of New Channel States in DTMC: To solve this problem, we want the states to evolve in different dimensions when channel state is different. For example, when the channel state changes, one chain stops evolving while the other starts evolving. More specifically, to generate a time reversible DTMC with a product form stationary distribution for a general topology where there are SUs in S (outside the interference range of the PU network), we introduce two Markov chains with new state definitions. We define[1]

$$\vec{y'}^a(t) = \{\vec{x'}(\tau): \text{ the largest } \tau \le t \text{ with } B(\tau) = 0\}$$

$$\vec{y'}^b(t) = \{\vec{x'}(\tau): \text{ the largest } \tau \le t \text{ with } B(\tau) = 1\}$$

Note that $\vec{y'}^a$ is the vector of transmission schedule in the most recent data slot (including time t) when the channel is ON; $\vec{y'}^b$ is the vector of transmission schedule in the most recent data slot (including time t) when the channel is OFF. We also denote the corresponding transmission schedules by y^a and y^b, respectively: $i \in y^a$ if and only if ${y'}_i^a = 1$ and $i \in y^b$ if and only if ${y'}_i^b = 1$. We then define $y^a = (\vec{y'}^a(t); B(t))$, $y^b = (\vec{y'}^b(t); B(t))$ as the states for the two chains, respectively. For instance, in Figure 5, two SUs are in the conflict set of each other with SU 1 in $\bar{S}$ and SU 2 in S: $(0,1;0)$ indicates that the channel is available and only SU link 2 is transmitting; $(1,0;1)$ means that the channel is unavailable and in the most recent available channel state, only SU link 1 is transmitting. The transitions in Figure 5 follows exactly from CA-CSMA described later in Section III-B. It is obvious that the transition to the next state depends only on the current state and the current input including the channel state B and the decision schedule. Hence both chains are Markovian. It is easy to verify, as in Figure 4, that Markov chains with y^a, y^b as the states are not time reversible.[2]

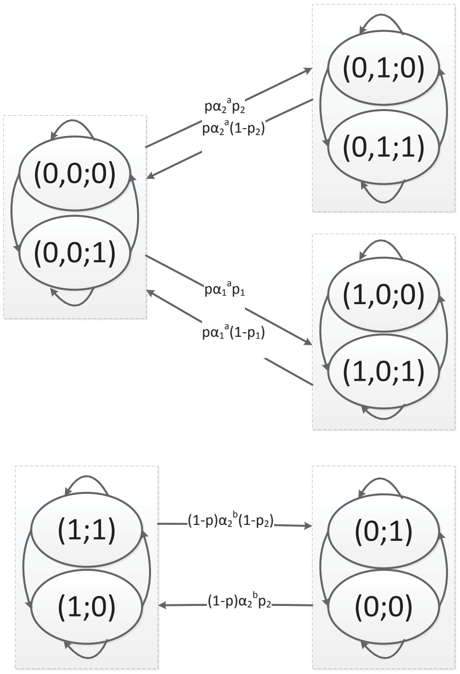

Fig. 5. Two DTMC evolutions for SUs inside and outside the interference range of the PU. Two SUs are in the conflict set of each other. Both DTMCs are time reversible and have product-form stationary distribution (by Propositions 3.2 and 3.4). α_2^a is the probability SU 2 is chosen the decision schedule m^a when the channel is available. α_2^b is the probability SU 2 is chosen in the decision schedule m^b when the channel is unavailable. Note that $|m^a| = 2$ and $|m^b| = 1$ in this example. The first DTMC transitions to another state only when the channel is ON while the second DTMC transitions to another state only when the channel is OFF.

We further define an aggregate state for each DTMC, which includes only $\vec{y}(t)$.

$$DTMC\ \bar{Y}^a : \bar{y}^a = (\vec{y'}^a(t))$$

$$DTMC\ \bar{Y}^b : \bar{y}^b = (\vec{y'}^b(t))$$

Note that in Figure 5, the rectangles in the first chain correspond to $\bar{y}^a$ and those in the second chain correspond to $\bar{y}^b$. In $DTMC\ \bar{Y}^a$, the transition to the next state $\hat{\bar{y}}^a$ depends only on the current state $\bar{y}^a$ and the current input including the channel state B and the decision schedule. Thus $DTMC\ \bar{Y}^a$ is Markovian. Similar arguments can be applied to $DTMC\ \bar{Y}^b$. In Section III-B, both DTMCs will be

[1] We assume t starts from $-\infty$ to make $\vec{y'}^a$, $\vec{y'}^b$ well-defined, especially for $\vec{y'}^a(t)$ when $B(0) = \cdots = B(t) = 1$ and $\vec{y'}^b(t)$ when $B(0) = \cdots = B(t) = 0$, respectively.

[2] Note that we also use Figure 5 to illustrate the Markov chains with the aggregate states $\bar{y}^a$ and $\bar{y}^b$ later. The transitioning probabilities for each single state are not labeled in Figure 5. By CA-CSMA, it is easy to find that the outgoing probabilities from $(\vec{y'}^a; 0)$ and $(\vec{y'}^a; 1)$ to $(\vec{\hat{y}'}^a; 0)$ are the same. The incoming probabilities from $(\vec{\hat{y}'}^a; B(t))$ to $(\vec{y'}^a; 1)$ do not exist if $\vec{\hat{y}'}^a \neq \vec{y'}^a$.

shown to be time reversible with a product form stationary distribution.

B. CA-CSMA: Scheduling Algorithm for Single Channel CRNs

Our new algorithm is based on the redefinition of the system state of Section III-A along with the differentiated treatment of SUs inside and outside the PU interference range. Each SU i keeps a queue denoted by q_i. The transmission schedule for q_i is denoted by x'_i. We define $A_i(t)$ as the arrivals to SU i at time slot t, $i = 1, \cdots, N$ and we assume it to be bounded. We assume arrivals are **i.i.d.** over time and independent between users. λ_n is defined to be $E(A_n(t))$. The evolution of the queue length is then written as:

$$q_i(t+1) = (q_i(t) + A_i(t) - z_i(t))^+, \text{ for } i = 1, \cdots, N,$$

where $z_i(t) = (1 - B(t))y'^a_i + B(t)y'^b_i(t)$. Note that $\vec{z}(t)$ only depends on $B(t)$, $\vec{y'}^a$ and $\vec{y'}^b$. Since we have shown that $\vec{y'}^a$ and $\vec{y'}^b$ are Markovian in Section III-A, the queue lengths at SUs evolve as a Markov Chain with the transitions caused by arrivals, departures and channel state in the current time slot. In Proposition 3.6, we will show that this Markov Chain is positive recurrent for any arrival rate vector within the actual capacity region (also called throughput optimal) under CA-CSMA.

1) Description of CA-CSMA: The decision schedule for all SUs is denoted by $m^a(t)$. The decision schedule for SUs in S, denoted by $m^b(t)$, does not consider the interference to or from SUs in $\bar{S}$. The set of all $m^a(t)$ and $m^b(t)$ are denoted by M^a_0 and M^b_0, respectively. We define $\alpha^a(m^a(t))$ and $\alpha^b(m^b(t))$ as the probability that $m^a(t)$ is chosen in the control slot when the channel is available, and the probability that $m^b(t)$ is chosen in the control slot when the channel is unavailable, respectively. *For clarification, "available" means the PU is idle and "unavailable" means the PU is active although SUs in S cannot sense it*. Recall that S is the set of all SUs that are outside the interference range of the PU network.

We first develop CA-CSMA that characterizes the different behaviors of SUs in S and $\bar{S}$ under different channel states. Algorithm 1 summarizes the proposed CA-CSMA. SUs in $\bar{S}$ acquire channel state information in every time slot by locally sensing the channel while SUs in S are notified by SUs in $\bar{S}$ for the new channel state. Next we elaborate on the behaviors of the SUs on the channel state transitions. When the channel state is available, all SUs treat the most recently available slot as their previous slot ignoring the unavailable period, and schedule packets in a way similar to Q-CSMA (Lines 2-11) where $\bar{p}_i = 1 - p_i$ [13]. When the channel state is unavailable, SU i in $\bar{S}$ remains silent and SUs in S treat the most recently unavailable slot as their previous slot ignoring the available period, and schedule packets in a way similar to Q-CSMA (Lines 13-22) [13]. In other words, SUs in $\bar{S}$ either retrieve or record information on the activities of SUs in $C(i)$ when channel state changes while SUs in S have to record and retrieve on the channel state change.

Algorithm 1 *CA-CSMA*

```
 1: if channel is available /*$B(t) = 0$*/ then
 2:     1. In the control slot, randomly select a decision schedule $m^a(t) \in M^a_0$ with probability $\alpha^a(m^a(t))$
 3:     if $i \in m^a(t)$ and $y'^a_j(t-1) = 0$ for all $j \in C(i)$ then
 4:         (a) $x'_i(t) = 1$ with probability $p_i$
 5:         (b) $x'_i(t) = 0$ with probability $\bar{p}_i$
 6:     else if $i \in m^a(t)$ and $y'^a_j(t-1) = 1$ for some $j \in C(i)$ then
 7:         $x'_i(t) = 0$
 8:     else
 9:         $x'_i(t) = y'^a_i(t-1)$ /*$i \notin m^a(t)$*/
10:     end if
11:     2. In the data slot, use $\vec{x'}(t)$ as the transmission schedule
12: else
13:     Execute Lines 2-11 by replacing all $a$ with $b$
14: end if
```

To understand CA-CSMA better, we consider the illustrative example (Figure 5) with SU 1 inside the interference range of the PU ($1 \in \bar{S}$) and SU 2 ($2 \in S$) outside. $(0, 1; 0)$ indicates that the channel is available and only SU 2 is transmitting; $(1, 0; 1)$ means the channel is unavailable and in the most recent available channel state, only SU 1 is transmitting. $(0; 1)$ indicates that the channel is unavailable and SU 2 is not transmitting; $(1; 0)$ means the channel is available and in the most recent unavailable channel state, SU 2 is transmit-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2343613, IEEE Transactions on Mobile Computing

ting. When the channel is available, the first chain in Figure 5 transitions to the next state though the second chain only stays in the previous state (Lines 2-11 in CA-CSMA); when the channel is unavailable, the second chain transitions to the next state while first chain only stays in the previous state (Lines 13-22).

Lines 2 and 13 in CA-CSMA can be implemented in a distributed manner through contention similar to [13] - the information exchange is kept locally. At time slot t: 1) SU i selects a random number T_i uniformly in $[0, W-1]$ and waits for T_i control mini-slots; 2) If SU i hears an INTENT message from a SU in $C(i)$ before the (T_i+1)-th control mini-slot, i will not be included in $m^a(t)$ or $m^b(t)$ and will give up the transmission of the INTENT message in this control slot; 3) If SU i does not hear an INTENT message from any SU in $C(i)$ before the (T_i+1)-th control mini-slot, it will broadcast an INTENT message at the beginning of the (T_i+1)-th control mini-slot. If there is no collision, SU i will be included in $m^a(t)$ or $m^b(t)$; or else, no SU is included. Note that we only need to send SU ID in the INTENT message. So even if the message is encoded with low transmission data rate at the physical layer, this overhead is insignificant. The other overhead of the algorithm includes notifications of channel state changes for SUs in S that could be done by SUs in $\bar{S}$. In this scheme, a control channel with limited bandwidth is used. On channel state change, each SU in $\bar{S}$ sends the notification to all its neighbors. SUs in $\bar{S}$ will silenty drop the message since they have already sensed the channel state change. Each SU in S forwards it to all its neighbors if a notification has not been received in this time slot. This scheme guarantees all SUs in $\bar{S}$ notified without infinite loops.

2) Proof of Optimality: In the following, we will formally show that CA-CSMA achieves throughput optimality. The transition probabilities are presented in Lemmas 3.1 and 3.3. Propositions 3.2 and 3.4 give the product-form of the stationary distribution. Proposition 3.6 claims the throughput optimality of CA-CSMA. We define $\pi(v) := prob$(state is v).

Lemma 3.1: (a) A state $\bar{y}^a = (\vec{y'}^a(t))$ can make a transition to a state $\bar{\hat{y}}^a = (\vec{\hat{y'}}^a(t))$ $(\vec{y'}^a \neq \vec{\hat{y'}}^a)$ iff

$$y^a \cup \hat{y}^a \in M_0^a$$

and there exists a decision schedule $m^a \in M_0^a$ s.t.

$$y^a \triangle \hat{y}^a := (y^a \setminus \hat{y}^a) \cup (\hat{y}^a \setminus y^a) \subseteq m^a.$$

(b) The transition probability $P(\bar{y}^a, \bar{\hat{y}}^a)$ from $\bar{y}^a$ to $\bar{\hat{y}}^a$ $(\neq \bar{y}^a)$.

$$\begin{aligned} & P(\bar{y}^a, \bar{\hat{y}}^a) \\ = & \sum_{m^a \in M_0^a : y^a \triangle \hat{y}^a \subseteq m^a} p\alpha^a(m^a) \Bigg(\prod_{l \in y^a \setminus \hat{y}^a} \bar{p}_l \Bigg) \\ & \Bigg(\prod_{k \in y^a \setminus \hat{y}^a} p_k \Bigg) \Bigg(\prod_{i \in m^a \cap (y^a \cap \hat{y}^a)} p_i \Bigg) \\ & \Bigg(\prod_{j \in m^a \setminus (y^a \cup \hat{y}^a) \setminus C(y^a \cup \hat{y}^a)} \bar{p}_j \Bigg), \end{aligned} \tag{1}$$

where $C(y^a \cup \hat{y}^a)$ denotes the neighbors of nodes in $y^a \cup \hat{y}^a$.

Proof: Part (a) can be proven as Lemma 2 in [13]. To prove Part (b), we denote $P^{sch}(\vec{y'}^a, \vec{\hat{y'}}^a)$ as $P(\vec{y'}, \vec{\hat{y'}})$ in Lemma 2 of [13] which is the transition probability from state $\vec{y'}$ to state $\vec{\hat{y'}}$ with the always-available channel. We only need to show

$$P((\vec{y'}^a; 0), \vec{\hat{y'}}^a) = P((\vec{y'}^a; 1), \vec{\hat{y'}}^a) = pP^{sch}(\vec{y'}^a, \vec{\hat{y'}}^a).$$

The first equality is obvious by CA-CSMA.

$$\begin{aligned} & P((\vec{y'}^a; 0), \vec{\hat{y'}}^a) \\ = & P((\vec{y'}^a; 0), (\vec{\hat{y'}}^a; 0)) + P((\vec{y'}^a; 0), (\vec{\hat{y'}}^a; 1)) \\ = & pP^{sch}(\vec{y'}^a, \vec{\hat{y'}}^a) + 0; \end{aligned}$$

$$\begin{aligned} & P((\vec{y'}^a; 1), \vec{\hat{y'}}^a) \\ = & P((\vec{y'}^a; 1), (\vec{\hat{y'}}^a; 0)) + P((\vec{y'}^a; 1), (\vec{\hat{y'}}^a; 1)) \\ = & pP^{sch}(\vec{y'}^a, \vec{\hat{y'}}^a) + 0. \end{aligned}$$

By Lemma 2 in [13] which states the transition probability with the always-available channel, we can prove Part (b). ■

Based on the state transition probabilities, we show that $DTMC^a$ has product-form stationary distributions and give the specific forms in Proposition 3.2. Since we have divided the state transitions in different channel states into two chains, either one of them can be treated as channel state unchanged, which simplifies our proof and leads to a clean result.

Proposition 3.2: A necessary and sufficient condition for the $DTMC^a$ to be irreducible and aperiodic is $\cup_{m^a \in M_0^a} m^a = \{1, \cdots, N\}$ and in this case the DTMC is reversible and has the following product-form stationary distribution: $\pi(\bar{Y}^a) = \frac{1}{Z^a} \prod\limits_{i \in y^a} \frac{p_i}{\bar{p_i}}$, $Z^a = \sum\limits_{y^a \in M_0^a} \prod\limits_{i \in y^a} \frac{p_i}{\bar{p_i}}$.

Proof: The necessary and sufficient condition can be proven as in the proof of Proposition 1 in [13] which states the product-form stationary distribution of the DTMC with the transmission schedule as the network state in an alway-available channel network. We only need to check $\pi(\bar{y}^a)P(\bar{y}^a, \bar{\hat{y}}^a) = \pi(\bar{\hat{y}}^a)P(\bar{\hat{y}}^a, \bar{y}^a)$. It follows that the DTMC is reversible and has such stationary distribution. ■

In a similar way, we show that $DTMC^b$ has product-form stationary distributions in Proposition 3.4 based on Lemma 3.3.

Lemma 3.3: (a) A state $\bar{y}^b = (\vec{y'}^b(t))$ can make a transition to a state $\bar{\hat{y}}^b = (\vec{\hat{y'}}^b(t))$ $(\vec{y'}^b \neq \vec{\hat{y'}}^b)$ iff

$$y^b \cup \hat{y}^b \in M_0^b$$

and there exists a decision schedule $m^b \in M_0^b$ s.t.

$$y^b \triangle \hat{y}^b := (y^b \setminus \hat{y}^b) \cup (\hat{y}^b \setminus y^b) \subseteq m^b.$$

(b) The transition probability $P(\bar{y}^b, \bar{\hat{y}}^b)$ from $\bar{y}^b$ to $\bar{\hat{y}}^b$ $(\neq \bar{y}^b)$.

$$\begin{aligned} & P(\bar{y}^b, \bar{\hat{y}}^b) \\ = & \sum_{m^b \in M_0^b : y^b \triangle \hat{y}^b \subseteq m^b} (1-p)\alpha^b(m^b) \left(\prod_{l \in y^b \setminus \hat{y}^b} \bar{p}_l \right) \\ & \left(\prod_{k \in y^b \setminus \hat{y}^b} p_k \right) \left(\prod_{i \in m^b \cap (y^b \cap \hat{y}^b)} p_i \right) \\ & \left(\prod_{j \in m^b \setminus (y^b \cup \hat{y}^b) \setminus C(y^b \cup \hat{y}^b)} \bar{p}_j \right), \end{aligned}$$

where $C(y^b \cup \hat{y}^b)$ denotes the neighbors of nodes in $y^b \cup \hat{y}^b$.

Proof: Part (a) can be proven as Lemma 2 in [13]. To prove Part (b), we denote $P^{sch}(\vec{y'}^b, \vec{\hat{y'}}^b)$ as $P(\vec{y'}, \vec{\hat{y'}}^b)$ in Lemma 2 of [13] which is the transition probability from state $\vec{y'}$ to state $\vec{\hat{y'}}$ with the always-available channel. We only need to show

$$\begin{aligned} P((\vec{y'}^b; 0), \vec{\hat{y'}}^b) &= P((\vec{y'}^b; 1), \vec{\hat{y'}}^b) \\ &= (1-p)P^{sch}(\vec{y'}^b, \vec{\hat{y'}}^b). \end{aligned}$$

The rest of the proof is similar to that of Lemma 3.1. ■

Proposition 3.4: A necessary and sufficient condition for the $DTMC^b$ to be irreducible and aperiodic is $\cup_{m^b \in M_0^b} m^b = S$ and in this case the DTMC is reversible and has the following product-form stationary distribution: $\pi(\bar{y}^b) = \frac{1}{Z^b} \prod\limits_{i \in y^b} \frac{p_i}{\bar{p_i}}$, $Z^b = \sum\limits_{y^b \in M_0^b} \prod\limits_{i \in y^b} \frac{p_i}{\bar{p_i}}$.

Proof: It is similar to the proof of Proposition 3.2 except that we need to check $\pi(\bar{y}^b)P(\bar{y}^b, \bar{\hat{y}}^b) = \pi(\bar{\hat{y}}^b)P(\bar{\hat{y}}^b, \bar{y}^b)$. ■

Based on the product-form distribution, we use the following results established in [12] to prove throughput-optimality of Algorithm 1.

Theorem 3.5: [12][3] We define $w^*(t) := \max\limits_{x \in \mathcal{M}(t)} \sum\limits_{i \in x} w_i(t)$ where $\mathcal{M}(t)$ is the set of all feasible schedules at time t. For a scheduling algorithm, if given any ϵ and δ, $0 < \epsilon, \delta < 1$, there exists a $\beta > 0$ such that: if $w^*(t) > \beta$, the scheduling algorithm chooses a schedule $x(t) \in \mathcal{M}(t)$ that satisfies

$$P\{\sum_{i \in x(t)} w_i(t) \geq (1-\epsilon)w^*(t)\} \geq 1 - \delta, \quad (2)$$

where $w_i(t) = f_i(q_i(t))$ is a function of queue lengths satisfying the following conditions:

1) $f_i(q_i(t))$ is a nondecreasing, continuous function with $\lim\limits_{q_i \to \infty} f_i(q_i) = \infty$;
2) Given any $a \in \mathbb{R}$, $\lim_{q_i \to \infty} \frac{f_i(q_i + a)}{f_i(q_i)} = 1$.

Then the scheduling algorithm is throughput optimal.

Remark: The throughput optimality result in Theorem 3.5 holds for any scheduler as long as the conditions are satisfied. It does not depend on how the scheduling algorithm is designed.

[3] The proof of the theorem in [12] can be easily extended to general conflict graph, and is applicable to our cognitive radio model. However, the authors in [12] can only find a scheduling algorithm that satisfies (2) in the fully-connected conflict graph while we propose a scheduling algorithm satisfying (2) for the general conflict graph under the cognitive radio framework.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2343613, IEEE Transactions on Mobile Computing

Then the scheduling algorithm is throughput-optimal.

We choose $p_i = \frac{e^{w_i(t)}}{e^{w_i(t)}+1}$ as long as w_i satisfies the conditions in [3]. By choosing f_i wisely, $w_i(t)$ evolves slowly over t. For instance, we choose $f_i(q_i) = \log(\log(q_i + e))$ in Section IV. We assume that the DTMC is in steady-state in every time slot throughout this paper (time-scale separation) [8][13]. In the following, we want to show that CA-CSMA is throughput-optimal by showing that it is "close" enough to another throughput-optimal scheduling algorithm - MWS. According to our design of two DTMCs, we analyze this "closeness" in different channel states separately, which is different from [13].

Proposition 3.6: Suppose $\cup_{m^a \in M_0^a} m = \{1, \cdots, N\}$ and $\cup_{m^b \in M_0^b} m^b = S$. Let $p_i = \frac{e^{w_i(t)}}{e^{w_i(t)}+1}$, $\forall i \in \{1, \cdots, N\}$ when $B(t) = 0$ and $p_i = \frac{e^{w_i(t)}}{e^{w_i(t)}+1}$, $\forall i \in S$ when $B(t) = 1$, where $w_i(t) = f_i(q_i(t))$ is a function of queue length satisfying the conditions established in Theorem 3.5. Then CA-CSMA is throughput-optimal.

Proof: By Propositions 3.2 and 3.4, we know that both DTMCs have product-form stationary distributions. Given any ϵ and δ s.t. $0 < \epsilon, \delta < 1$. For $DTMC^a$, we define $w^a(t) = \max_{x \in M_0^a} \sum_{i \in x} w_i(t)$. Based on this, four sets of states are defined as follows:

$$\chi_0^a := \{(\vec{y'}^a; 0) |\ y^a \in M_0^a, \sum_{i \in y^a} w_i(t) < (1-\epsilon) w^a(t)\}$$

$$\chi_1^a := \{(\vec{y'}^a; 1) |\ y^a \in M_0^a, \sum_{i \in y^a} w_i(t) < (1-\epsilon) w^a(t)\}$$

$$\varphi^a := \chi_0^a \cup \chi_1^a$$

$$\begin{aligned} \psi^a \quad := \quad & \{\bar{Y}^a = (\vec{y'}^a) |\ y^a \in M_0^a, \\ & \sum_{i \in y^a} w_i(t) < (1-\epsilon) w^a(t), \} \end{aligned}$$

where χ_0^a includes all states with the channel available and the sum of $w_i(t)$ from SUs chosen in the schedule is at least a fraction of ϵ away from $w^a(t)$, χ_1^a includes all states with the channel unavailable and the sum of $w_i(t)$ from SUs chosen in the schedule of the most recently available slot is at least a fraction of ϵ away from $w^a(t)$. Note that if $(\vec{y'}^a; B) \in \varphi^a$, then $\bar{y}^a = (\vec{y'}^a) \in \psi^a$. We then calculate the probability of a state in set χ_0^a. We define $\pi(A) := prob$ (state $v \in A$).

$$\begin{aligned} & \pi(\chi_0^a) < \pi(\varphi^a) = \pi(\psi^a) \\ = \quad & \sum_{\vec{y'}^a \in \psi^a} \pi(\bar{Y}^a) = \sum_{\vec{y'}^a \in \psi^a} \frac{e^{\sum\limits_{i \in y^a} w_i(t)}}{Z^a} \\ \leq \quad & \frac{|\psi^a| e^{(1-\epsilon) w^a(t)}}{Z^a} < \frac{2^N}{e^{\epsilon w^a(t)}} \end{aligned}$$

where

$$Z^a = \sum_{y^a \in M_0^a} e^{\sum\limits_{i \in y^a} w_i(t)} > e^{\max\limits_{y^a \in M_0^a} \sum\limits_{i \in y^a} w_i(t)} = e^{w^a(t)}.$$

The first equality holds because $1_{\{(\vec{y'}^a; 0)\} \cup \{(\vec{y'}^a; 1)\}} = 1_{\{(\vec{y'}^a)\}}$; The last inequality is true because $|\psi^a| \leq |M_0^a| \leq 2^N$. Thus, $\exists \beta^a > 0$, such that: $w^a(t) > \beta^a$ implies $\pi(\chi_0^a) < \delta \min(p, 1-p)$.

For $DTMC^b$, we define $w^b(t) = \max_{x \in M_0^b} \sum_{i \in x} w_i(t)$. Similar to $DTMC^a$, four sets of states are defined.

$$\chi_0^b := \{(\vec{y'}^b; 0) |\ y^b \in M_0^b, \sum_{i \in y^b} w_i(t) < (1-\epsilon) w^b(t)\}$$

$$\chi_1^b := \{(\vec{y'}^b; 1) |\ y^b \in M_0^b, \sum_{i \in y^b} w_i(t) < (1-\epsilon) w^b(t)\}$$

$$\varphi^b := \chi_0^b \cup \chi_1^b$$

$$\begin{aligned} \psi^b \quad := \quad & \{\bar{Y}^b = (\vec{y'}^b) |\ y^b \in M_0^b, \\ & \sum_{i \in y^b} w_i(t) < (1-\epsilon) w^b(t)\} \end{aligned}$$

We then calculate the probability of a state in set χ_1^b.

$$\begin{aligned} & \pi(\chi_1^b) < \pi(\varphi^b) = \pi(\psi^b) \\ = \quad & \sum_{\vec{y'}^b \in \psi^b} \pi(\bar{y}^b) = \sum_{\vec{y'}^b \in \psi^b} \frac{e^{\sum\limits_{i \in y^b} w_i(t)}}{Z^b} \\ \leq \quad & \frac{|\psi^b| e^{(1-\epsilon) w^b(t)}}{Z^b} < \frac{2^{|S|}}{e^{\epsilon w^b(t)}} \end{aligned}$$

where

$$Z^b = \sum_{y^b \in M_0^b} e^{\sum\limits_{i \in y^b} w_i(t)} > e^{\max\limits_{y^b \in M_0^b} \sum\limits_{i \in y^b} w_i(t)} = e^{w^b(t)}.$$

The first equality holds because $1_{\{(\vec{y'}^b;0)\} \cup \{(\vec{y'}^b;1)\}} = 1_{\{(\vec{y'}^b)\}}$; The last inequality is true because $|\psi^b| \leq |M_0^b| \leq 2^{|S|}$. Thus, $\exists \beta^b > 0$, such that: $w^b(t) > \beta^b$, which implies that $\pi(\chi_1^b) < \delta \min(p, 1-p)$.

Since $w^*(t) = w^a(t)$ when $B(t) = 0$, $w^*(t) = w^b(t)$ when $B(t) = 1$, we have the following results:

$$\begin{aligned}
& P\{\sum_{i \in y^a(t)} w_i(t) \geq (1-\epsilon) w^*(t) | B(t) = 0\} \\
= \ & P\{\sum_{i \in y^a(t)} w_i(t) \geq (1-\epsilon) w^a(t) | B(t) = 0\} \\
= \ & 1 - P\{\sum_{i \in y^a(t)} w_i(t) < (1-\epsilon) w^a(t) | B(t) = 0\} \\
= \ & 1 - \frac{\pi(\chi_0^a)}{p} > 1 - \delta \min(p, 1-p)/p \\
\geq \ & 1 - \delta \qquad (3)
\end{aligned}$$

when $w^a(t) > \beta^a$. Equation (3) implies that: If $B(t) = 0$, $P\{\sum\limits_{i \in y^a(t)} w_i(t) \geq (1-\epsilon) w^*(t)\} > 1 - \delta$ if $w^a(t) > \beta$. Similarly,

$$\begin{aligned}
& P\{\sum_{i \in y^b(t)} w_i(t) \geq (1-\epsilon) w^*(t) | B(t) = 1\} \\
= \ & P\{\sum_{i \in y^b(t)} w_i(t) \geq (1-\epsilon) w^b(t) | B(t) = 1\} \\
= \ & 1 - P\{\sum_{i \in y^a(t)} w_i(t) < (1-\epsilon) w^a(t) | B(t) = 1\} \\
= \ & 1 - \frac{\pi(\chi_1^b)}{1-p} > 1 - \delta \min(p, 1-p)/(1-p) \\
\geq \ & 1 - \delta \qquad (4)
\end{aligned}$$

when $w^b(t) > \beta^b$. Equation (4) implies that: If $B(t) = 1$, $P\{\sum\limits_{i \in y^b(t)} w_i(t) \geq (1-\epsilon) w^*(t)\} > 1 - \delta$ when $w^b(t) > \beta$.

We use the total probability formula to calculate the unconditional probability:

$$\begin{aligned}
& P\{\sum_{i \in x(t)} w_i(t) \geq (1-\epsilon) w^*(t)\} \\
\overset{(a)}{=} \ & P\{\sum_{i \in y^a(t)} w_i(t) \geq (1-\epsilon) w^*(t) | B(t) = 0\} \\
& P(B(t) = 0) \\
+ \ & P\{\sum_{i \in y^b(t)} w_i(t) \geq (1-\epsilon) w^*(t) | B(t) = 1\} \\
& P(B(t) = 1) \\
\overset{(b)}{\geq} \ & (1-\delta)p + (1-\delta)(1-p) = 1 - \delta
\end{aligned}$$

if $w^*(t) > \beta$ where $\beta = \max(\beta^a, \beta^b)$. Note that (a) holds because $y^a(t) = x(t)$ when $B(t) = 0$ and $y^b(t) = x(t)$ when $B(t) = 1$ by definition, and (b) holds due to Equations (3) and (4). Hence Algorithm 1 satisfies the condition of Theorem 3.5 and is throughput optimal. Although the scheduler in CA-CSMA involves the evolutions of two Markov chains where a combination of the transmission schedule and the channel state is the system state, it is throughput optimal as long as it is Markovian and the conditions in Theorem 3.5 are satisfied. ■

Proposition 3.6 shows that the full capacity region can be achieved by CA-CSMA in CRNs with SUs inside and outside the interference range of the PU. Similar to Lemma 3 in [13], we can prove that both $m^a(t)$ and $m^b(t)$ produced by this distributed implementation are feasible schedules and they satisfy the conditions in Proposition 3.6.

IV. SIMULATIONS

In this section, we conduct simulations to compare the performance of 802.11 (we use a similar algorithm as in [13]), MWS, Q-CSMA and CA-CSMA. Since the 802.11 algorithm we compare does not follow the exact designs of contention window sizes, sensing slots, transmission slots, etc, we call it "simple 802.11" in all the figures. In Q-CSMA, if a link is activated but the channel is off for it, it will keep silent. In CA-CSMA, W, the contention window size in the control slot, is chosen to be the number of SUs in the network. SU weights are chosen to be of the form $\log(\log(q+e))$, where q is the queue length [13] [14] [4]. We set $p = 0.6$, that is, 60% of the time, the PU is not using the channel. Note that in all the performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2343613, IEEE Transactions on Mobile Computing

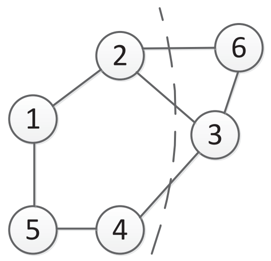

Fig. 6. Conflict graph with 6 SUs.

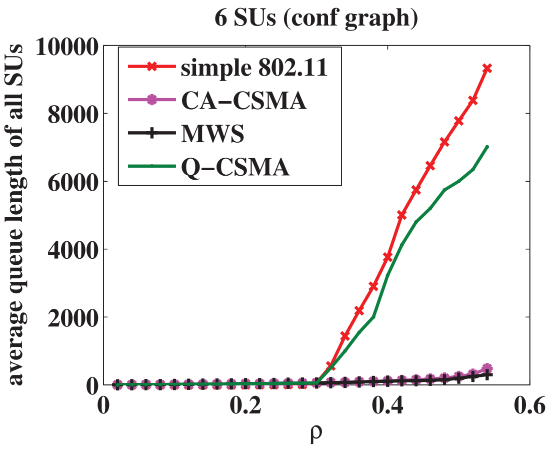

Fig. 7. Queue lengths of four algorithms with different loads in the 6-SU network.

evaluations, we do not take into account the control slot overhead since it is a fixed and small portion of the whole time slot. Discounted by this fixed factor, the performance of CA-CSMA is still promising as we will observe in the following evaluations.

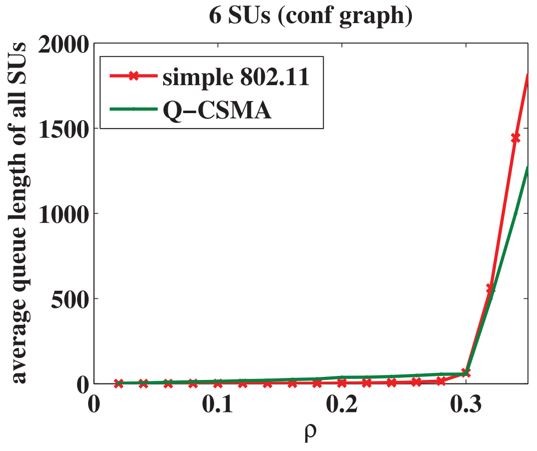

Fig. 8. Queue lengths of simple 802.11 and Q-CSMA with low loads in the 6-SU network.

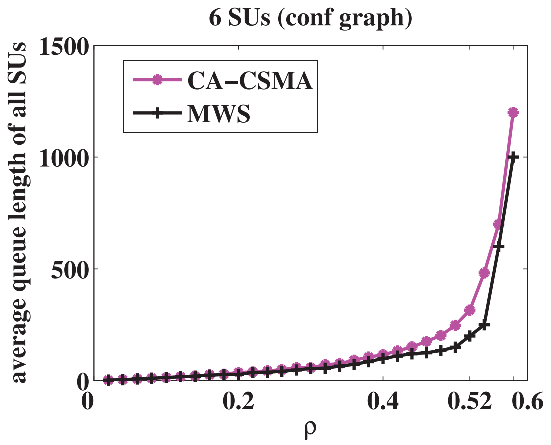

Fig. 9. Queue lengths of CA-CSMA and MWS with a wide load range in the 6-SU network.

A. Performance Evaluation in the 6-SU Network

In “Network 1”, there are 6 SUs whose conflict graph is shown in Figure 6. A conflict graph is one where two SUs are neighbors if they cannot transmit simultaneously. SUs 3 and 6 are outside the interference range of the PU and others are inside. Let $\lambda = 0.2\times(1,0,1,0,0,0)+0.3\times(1,0,0,1,0,1)+0.2 \times (0,1,0,0,1,0) + 0.3 \times (0,0,1,0,1,0) = (0.5,0.2,0.5,0.3,0.5,0.3)$, which is a convex combination of some maximal independent sets. We vary ρ from 0 to $0.9 \times p$ so that $\rho \times \lambda$ lies inside the capacity region. For each algorithm, for a fixed ρ, we run 10 independent experiments and take the average. We show the average queue length of the network over ρ in Figure 7 where the running time is 10^5 time slots. As we can see, CA-CSMA outperforms simple 802.11, which does not take into account the queue length information, and Q-CSMA, which ignores channel state information. We plot Figure 8 to further show the rate regions of simple 802.11 and Q-CSMA. The blow up point is observed to be at around $\rho = 0.3$. Accordingly, we believe the improvement of CA-CSMA comes from Markov chain separation instead of pure queue length based algorithm. In addition, CA-CSMA performs almost as well as MWS. Figure 9 extends ρ to 0.58, which pushes the load almost to the capacity region given that $p = 0.6$, to show the full performances of CA-CSMA and MWS. MWS is slightly slower than CA-CSMA in terms of queue length growth over a wide range of traffic load. The rapid increase of queue lengths of both algorithms occurs at around $\rho = 0.52$. In [13], a hybrid

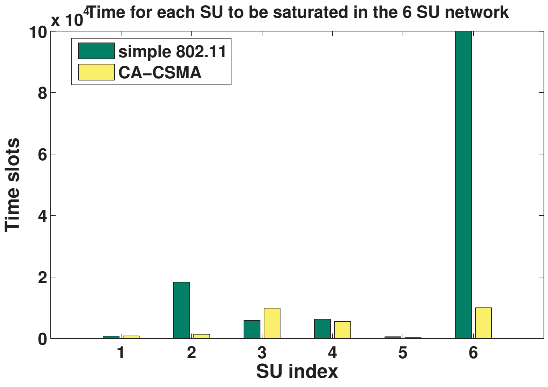

Fig. 10. Time for each SU to saturate in the 6-SU network.

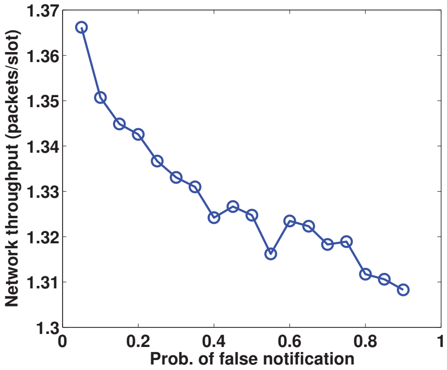

Fig. 11. Network throughput over the probability of false notification of channel state in the 6-SU network.

algorithm is developed to reduce the delay while maintaining the property of throughput optimality. A hybrid algorithm based on CA-CSMA can be similarly designed to further improve the delay performance, which is not the focus of this paper.

To see how long it takes the algorithms to saturate the network, we define a queue length threshold, above which the SU is saturated. This threshold is set to be $500/6 \doteq 83$ in the 6-SU network. Note that 500 is the average number of packets in the network reached by CA-CSMA as shown in Figure 7. We plot the time for queue length of each SU to reach this threshold using both simple 802.11 and CA-CSMA in Figure 10. The simulation time is 10^5 times slot. If the queue length does not reach 83 packets within the simulation time, we set the saturation time to be 10^5. The saturation times at SUs under CA-CSMA are much more balanced than in simple 802.11, which may also explain why CA-CSMA stabilizes the system with arrivals that simple 802.11 cannot.

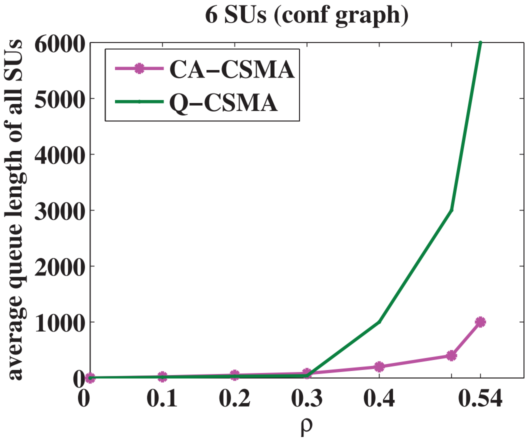

Fig. 12. Queue lengths of CA-CSMA and Q-CSMA with slowly varying channel state.

In CA-CSMA, channel state changes need to be notified to the SUs outside the interference range of the PU, which is an extra overhead and may cause errors as well. In Figure 11, we show the robustness of the network throughput to notification errors. We assume that the wrong notification in the previous slot will be corrected in the current slot, which means, SUs outside the interference range of the PU would know they got the wrong notification in the previous slot. This assumption allows the design of an efficient correction algorithm on the notification errors, which is also our future work. The network throughput degrades gracefully when the error increases in Figure 11.

We also evaluate the performances of CA-CSMA and Q-CSMA in a more practical setting - lower channel variation, i.e., a common scenario of cognitive radio on TV band. In the first time slot, the channel state is available with probability of p. We then force the channel state to be unchanged for the next 9 time slots. This process is repeated with a period of 10 time slots. We observe that CA-CSMA still outperforms Q-CSMA although the queue length of the latter grows more slowly than in Figure 7. The performance gain is believed to come from the slower channel state change. The positive results of CA-CSMA leads us to our future work on proving its throughput-optimality without i.i.d. assumption in PU activity.

B. *Performance Evaluation in the 16-SU Network*

There are 16 SUs in “Network 2”, where the conflict graph is a 4 by 4 grid (Figure 13). Note that

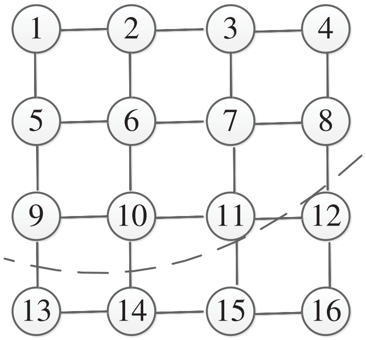

Fig. 13. Conflict graph with 16 SUs.

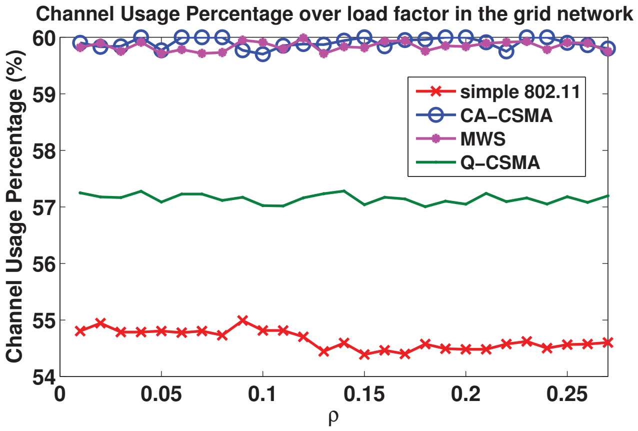

Fig. 14. Channel usage percentage over load factor in the 4 by 4 grid network.

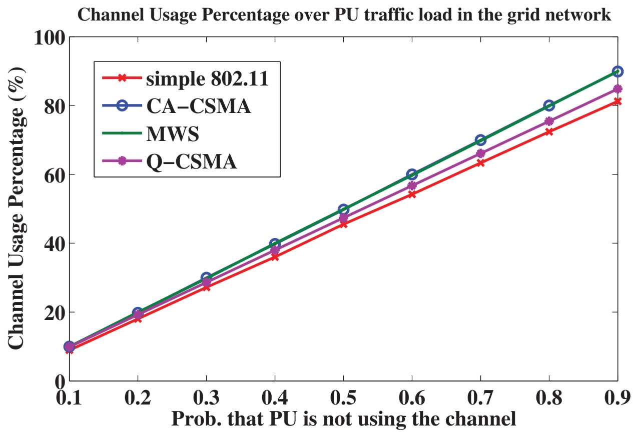

Fig. 15. Channel usage percentage over PU traffic in the 4 by 4 grid network.

the sizes we choose in the simulations are comparable to those in [13] and [8]. SUs 1 through 11 are within the interference range of the PU, while SUs 12 to 16 are not. We compare metrics specific in cognitive radio networks for simple 802.11, MWS, Q-CSMA and CA-CSMA in “Network 2”. First, we define channel usage as the percentage that any of the SUs in the interference range of the PU is using the channel. In Figure 14, we vary the load factors from 0 to close to capacity region boundary and compare the channel usage percentages. All algorithms fluctuate but there are no significant drops or jumps. Both MWS and CA-CSMA utilize the available channel bandwidth efficiently while Q-CSMA and simple 802.11 show worse channel usage.

We compare channel usage over different PU traffic loads in Figure 15 where the SU traffic load is 90% of the capacity region (in the comparisons of all the following metrics, we use the same load factor if not specifically mentioned). Similarly, CA-CSMA and MWS utilize the channel bandwidth better than Q-CSMA and simple 802.11 in all PU traffic loads.

V. Conclusion

In this paper, we develop CA-CSMA: a throughput optimal distributed queue length based CSMA/CA scheduling algorithm for cognitive radio networks. The algorithm needs to signal a group of SUs during channel state change, different from existing distributed queue-length based CSMA/CA algorithms. Our algorithm is designed to adapt to the channel availability dynamics caused by unknown PU activity. The performance of CA-CSMA, MWS, Q-CSMA and a simplified 802.11 are compared in simulations to show the efficacy of CA-CSMA.

In the future, we plan to relax the time-scale separation assumption, as in [8], [14], and extend our work to multi-hop and multi-PU cognitive radio networks. Another topic of future interest is to characterize the impact of sensing errors on algorithm design.

References

[1] Prasanna Chaporkar, Koushik Kar, and Saswati Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *In Proceedings of 43d Annual Allerton Conference on Communication, Control and Computing*, pages 28–30, 2005.

[2] Antonis Dimakis and Jean Walrand. Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits. *Advances in Applied Probability*, 38(2):pp. 505–521, 2006.

[3] Atilla Eryilmaz, Rayadurgam Srikant, and James R. Perkins. Stable scheduling policies for fading wireless channels. *IEEE/ACM Transactions on Networking*, 13:411–424, April 2005.

[4] Javad Ghaderi and Rayadurgam Srikant. On the design of efficient csma algorithms for wireless networks. In *Proc. IEEE conference on Decision and Control*, December 2010.

[5] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388 –404, March 2000.

[6] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201 – 220, February 2005.

[7] http://grouper.ieee.org/groups/802/22/. Ieee 802.22, working group on wireless regional area networks (wran).

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

[9] Changhee Joo, Xiaojun Lin, and N.B. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1103 –1111, April 2008.

[10] F. P. Kelly. *Reversibility and Stochastic Networks*. Cambridge University Press, New York, NY, USA, 2011.

[11] Mathieu Leconte, Jian Ni, and Rayadurgam Srikant. Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 165–174, New York, NY, USA, 2009. ACM.

[12] Bin Li and Atilla Eryilmaz. A fast-csma algorithm for deadline constraint scheduling over wireless fading channels. In *Workshop on Resource Allocation and Cooperation in Wireless Networks (RAWNET)*, May 2011.

[13] Jian Ni, Bo 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 the 29th conference on Information communications*, INFOCOM'10, pages 271–275, Piscataway, NJ, USA, 2010. IEEE Press.

[14] Shreevatsa Rajagopalan, Devavrat Shah, and Jinwoo Shin. Network adiabatic theorem: an efficient randomized protocol for contention resolution. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, SIGMETRICS '09, pages 133–144, New York, NY, USA, 2009. ACM.

[15] F.W. Seelig. A description of the august 2006 xg demonstrations at fort a.p. hill. In *DySPAN 2007. 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2007*, pages 1 –12, April 2007.

[16] 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*, 37(12):1936 –1948, December 1992.

[17] Rahul Urgaonkar and Michael J. Neely. Opportunistic scheduling with reliability guarantees in cognitive radio networks. *IEEE Transactions on Mobile Computing*, 8:766–777, 2009.

[18] Dongyue Xue and Eylem Ekici. Guaranteed opportunistic scheduling in multi-hop cognitive radio networks. In *INFOCOM 2011. The 30th Conference on Computer Communications. IEEE*, April 2011.

Shuang Li received her Ph.D in Computer Science from The Ohio State University in 2013, M.S. in Computer Science from Auburn University in 2008, and B.E. in Computer Science from Wuhan University of Technology, China in 2005. .

She is currently a software engineer at Google Fiber. Her research interests include resource allocation and network optimization in wireless networks, especially in cognitive radio networks, and high performance routing and switching in optical networks.

Eylem Ekici received the B.S. and M.S. degrees in computer engineering from Bogazici University, Istanbul, Turkey, in 1997 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2002. Currently, he is an Associate Professor with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus. His current research interests include cognitive radio networks, wireless sensor networks, vehicular communication systems, and nano communication systems with a focus on modeling, optimization, resource management, and analysis of network architectures and protocols. Dr. Ekici is an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE Transactions on Mobile Computing, Computer Networks, and Mobile Computing and Communications Review.

Ness Shroff received his Ph.D. degree in Electrical Engineering from Columbia University in 1994. He joined Purdue university immediately thereafter as an Assistant Professor in the school of ECE. At Purdue, he became Full Professor of ECE in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined The Ohio State University, where he holds the Ohio Eminent Scholar endowed chair in Networking and Communications, in the departments of ECE and CSE. From 2009-2012, he served as a Guest Chaired professor of Wireless Communications at Tsinghua University, Beijing, China, and currently holds an honorary Guest professor at Shanghai Jiaotong University in China. His research interests span the areas of communication, social, and cyberphysical networks. He is especially interested in fundamental problems in the design, control, performance, pricing, and security of these networks. Dr. Shroff is a past editor for IEEE/ACM Trans. on Networking and the IEEE Communication Letters. He currently serves on the editorial board of the Computer Networks Journal, IEEE Network Magazine, and the Networking Science journal. He has chaired various conferences and workshops, and co-organized two workshops for the NSF to chart the future of communication networks. Dr. Shroff is a Fellow of the IEEE and an NSF CAREER awardee. His work has received numerous best paper awards for his research, e.g., at IEEE INFOCOM 2008, IEEE INFOCOM 2006, Journal of Communication and Networking 2005, and Computer Networks 2003 (his papers also received runner-up awards at IEEE INFOCOM 2005 and IEEE INFOCOM 2013), and also student best paper awards (from all papers whose first author is a student) at IEEE WiOPT 2013, IEEE WiOPT 2012, and IEEE IWQoS 2006.