

# Optimal CSMA: A Survey

(Invited Paper)

Se-Young Yun, Yung Yi

Dept. of EE, KAIST, Daejeon, Korea

Email: yunseyoung@kaist.ac.kr,  
yiyung@kaist.edu

Jinwoo Shin

School of Computer Science,  
Georgia Institute of Technology, USA  
Email: jshin72@cc.gatech.edu

Do Young Eun

Dept. of ECE, North Carolina State Univ.,

Raleigh, NC, USA  
Email: dyeun@ncsu.edu

**Abstract**—Carrier Sense Multiple Access (CSMA) has been widely used as a medium access control (MAC) scheme in wireless networks mainly due to its simple and totally distributed operations. Recently, it has been reported in the community that even such simple CSMA-type algorithms can achieve optimality in terms of throughput and utility, by smartly controlling its operational parameters such as backoff and holding times. In this survey paper, we summarize the recent research efforts in this area with main focus on the key intuitions and rationales, and conclude by presenting some open problems.

## I. INTRODUCTION

Carrier Sense Multiple Access (CSMA) is one of most popular random access protocols in practice, which we see in most of wireless textbooks. The key features of CSMA is that each link with a pair of transmitter and receiver senses the medium and transmits a packet only if the medium is sensed idle. Due to its simple and distributed nature, it has been regarded as one of the most practical MAC protocols in wireless networks, e.g., CSMA is a basic medium access algorithm in IEEE 802.11. Thus, there exists a vast array of research results on CSMA in terms of its analysis under various settings and its applications to practical systems.

CSMA is referred to as the class of algorithms to schedule links over time in wireless networks. There are also numerous other types of algorithms in the area of wireless link scheduling, where their performances are often measured by various metrics, e.g. throughput, delay, fairness, etc. It's the year 1992 that a seminal paper by Tassiulas and Ephremides [1] was published, in which the so-called throughput optimality was defined, and a scheduling algorithm achieving throughput optimality, referred to as *Max-Weight*, was presented. Despite its provable optimality, *Max-Weight* requires to solve a computationally intractable problem, called Maximum Weight Independent Set problem, over each time, which has been a major obstacle to implement it in practice.

Since the work on *Max-Weight*, a surge of papers on MAC scheduling, which essentially follows the philosophy of *Max-Weight*, have been published. They partially or fully guarantee the performance, typically in terms of throughput and utility, where the efforts have been classified into (i) ones which trade off between complexity and efficiency, (ii) ones which achieve optimality at the cost of increasing delay, and (iii) random access style algorithms with suboptimality but worst-case performance (e.g., lower bound of the performance) guarantee, see e.g., [2] for a survey. A single sentence summary of the key ideas of all the above-mentioned research would be: Balancing the supply-demand differential by prioritizing links with larger differentials in scheduling algorithms, where differentials are quantified by the link queue lengths.

However, many aforementioned algorithms still require heavy message passing or computations, thus remain just

"theoretical" rather than being made "practical". Therefore, it has been a long-standing open problem to find simple random access (hopefully, without message passing) achieving *full* optimality in the research community. About 15 years after *Max-Weight*, it's the year 2008 that a simple CSMA with no message passing was shown to be provably optimal in terms of throughput and utility. Since then more and more research interests in this so-called *optimal CSMA* area have been taken in the community, whose survey is the major content of this paper. For convenience, we survey the research results on optimal CSMA based on the following criteria reflecting different models, proof techniques, and research methodologies (e.g., theory or implementation).

- C1. Saturate vs. Unsaturated.** In unsaturated cases, there is arrival of traffic with finite workload to each link, and stability is a key metric, whereas in saturated cases, there is infinite backlog behind each link, and utility value of the equilibrium rate is often the objective to be maximized.
- C2. Synchronous vs. Asynchronous.** Synchronous systems have a notion of *frames*, each of which typically consists of a control phase and a data phase, where frames are synchronized, whereas in asynchronous systems, each link independently accesses the medium after sensing other links' transmissions.
- C3. Continuous vs. Discrete.** This criterion can also be called with vs. without collisions. For mathematical tractability, continuous models are often used, where backoff and holding times can be arbitrary real numbers. In practice, the systems are actually discrete, where the systems evolve over discretized time slots (e.g., 20  $\mu$ sec in IEEE 802.11b) and collisions will inevitably occur, when two links contend at a same time slot.
- C4. Time-varying channels vs. static channels.** Static channels are often assumed mainly for analytical simplicity, where every link capacity is set fixed. Wireless channels, however, are time-varying in practice, where the results on optimal CSMA may significantly change, depending on the time-scale difference between the speed of channel variations and CSMA parameter controls.
- C5. Time-scale separation vs. not.** As will be discussed later in more detail, the behavior of optimal CSMA is modeled by a Markov chain, and this time-scale separation assumption corresponds to whether the Markov chain reaches a stationary distribution immediately or not. Results based on this "fake" assumption have been accepted in the community without much criticism, especially when analyzing the CSMA Markov chain becomes mathematically intractable.

TABLE I  
TAXONOMY OF OPTIMAL CSMA. TSS: TIME-SCALE SEPARATION

	Work	Sat/Unsat	Cont/Disc	Sync/Async	w/w.o. TSS	Summary and Comments
Theoretical works	[3]	Unsat	Cont	Async	With	A CSMA that is conjectured to be optimal
	[4]	Unsat	Cont	Async	Without	Complete proof of [3]
	[5]	Unsat	Disc	Async	Without	Throughput optimal with collision
	[6]	Unsat	Cont	Async	Without	Queue based approach with full optimality proof without TSS
	[7]	Unsat	Disc	Async	Without	Connecting Max-weight and CSMA with maximum queue size estimation
	[8]	Unsat	Cont	Async	Without	Continuous time version of [7]
	[9]	Sat	Cont	Async	Without	Utility optimal CSMA based on stochastic approximation with Markovian noise
	[10]	Sat	Cont	Async	Without	Utility optimal CSMA under multiple channels
	[11]	Unsat	Disc	Sync	With	Queue based approach under synchronous system
	[12]	Unsat	Disc	Sync	Without	Bounding delay based on parallel update of transmission aggressiveness
	[13]	Unsat	Disc	Async	With	Throughput optimal for imperfect carrier sensing
	[14]	Unsat	Cont	Async	Without	Delay of optimal CSMA algorithms based on asymptotic variance
	[15]	Unsat	Cont	Async	With	MIMO and SINR-based interference model
	[16], [17]	Unsat	Cont	Async	Without	CSMA over time-varying channel
	Impl.	[18], [19]		Disc	Async	
[20], [21]			Disc	Async		Study of interaction between CSMA and TCP
[22]			Disc	Async		A new MAC and experimental validation on 802.11 hardware

**C6. Theory vs. implementation.** Most of works in the literature have produced theoretical results with emphasis on discovering CSMA's ability toward optimality. There are also some of recent researches which implement and evaluate optimal CSMA, in conjunction with several redesign proposals to bridge the gap between theory and practice.

Following these six criteria, we summarize the key features of the research papers on optimal CSMA in Table I. The rest of the paper is devoted to explaining their key concepts and brief summaries.

## II. CSMA: MODEL AND OBJECTIVES

### A. Model

In wireless networks, each link shares the wireless medium with other neighbor links that interfere with the link. To model this, a wireless network topology is represented as an interference graph, where links are vertices and undirected edges are generated between two interfering links. Let  $G = (\mathcal{L}, E)$  denote the interference graph, where  $\mathcal{L}$  and  $E$  are the set of links and the set of edges between interfering links, respectively. We define by  $\sigma \triangleq [\sigma_i : i \in \mathcal{L}]$ <sup>1</sup> a scheduling vector for links in  $G$ . Since interfering links cannot successfully transmit a packet simultaneously,  $\sigma$  is called feasible (i.e., there is no collision) if  $\sigma_i + \sigma_j \leq 1, \forall (i, j) \in E$ , where  $(i, j)$  denotes the edge between link  $i$  and  $j$ . Thus, the set of all feasible schedules is defined as

$$\mathcal{I}(G) \triangleq \{\sigma \in \{0, 1\}^n : \sigma_i + \sigma_j \leq 1, \forall (i, j) \in E\}.$$

<sup>1</sup>Let  $[x_i : i \in \mathcal{L}]$  denote the vector whose  $i$ -th element is  $x_i$ . For notational convenience, instead of  $[x_i : i \in \mathcal{L}]$ , we use  $[x_i]$  in the remaining of this paper.

The feasible rate region (or capacity)  $C = C(G)$  is convex hull of  $\mathcal{I}(G)$ , namely,

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

Under CSMA, prior to trying to transmit a packet, links check whether the medium is busy or idle, and transmit the packet only when the medium is sensed idle. To control the aggressiveness of medium access, a notion of backoff timer is used, which is reset to a random value when it expires. The timer ticks only when the medium is idle. With the backoff timer, links try to avoid collisions by the following procedure: each link does not start transmission immediately when the medium is sensed idle, but keeps silent until its backoff timer expires. After a link grabs the channel, the link holds the channel for some duration, called holding time. Intuitively, the probability that link  $i$  is scheduled is decided by the average backoff time and the average holding time. Let the backoff and holding times be denoted by  $1/b_i$  and  $h_i$ , respectively.

For tractability, if we assume that backoff and holding times follow memoryless (i.e., exponential) distributions, the scheduling process  $\{\sigma(t)\}$  of CSMA protocols becomes a time reversible Markov process. Then, the stationary distribution of a schedule  $\sigma$  is defined by  $\mathbf{b} = [b_i]$  and  $\mathbf{h} = [h_i]$ :

$$\pi_{\sigma}^{\mathbf{b}, \mathbf{h}} = \frac{\prod_{i \in \mathcal{L}} (b_i h_i)^{\sigma_i}}{\sum_{\sigma' \in \mathcal{I}(G)} \prod_{i \in \mathcal{L}} (b_i h_i)^{\sigma'_i}}, \quad (1)$$

which is a function of the product  $b_i \times h_i$ , for all  $i \in \mathcal{L}$ . Let  $r_i = \log(b_i h_i)$  and  $\mathbf{r} = [r_i]$ , where  $\mathbf{r}$  implicitly denotes transmission aggressiveness of links. From (1), the probability  $s_i(\mathbf{r})$  that link  $i$  is scheduled for  $\mathbf{r}$ , which is the link  $i$ 's

throughput, is computed as follows:

$$s_i(\mathbf{r}) = \sum_{\sigma \in \mathcal{I}(G): \sigma_i=1} \pi_{\sigma}^{\mathbf{b}, \mathbf{h}} = \frac{\sum_{\sigma \in \mathcal{I}(G): \sigma_i=1} \exp(\sum_{i \in \mathcal{L}} \sigma_i r_i)}{\sum_{\sigma' \in \mathcal{I}(G)} \exp(\sum_{i \in \mathcal{L}} \sigma'_i r_i)}.$$

In the discrete time model, where geometric distributions are used for backoff and holding time instead of exponential, due to collisions, the stationary distribution is slightly different from (1). However, the stationary distribution becomes close to (1) when the holding time  $\mathbf{h}$  is large enough so that the collision time become ignorable, since the time fraction of collision period declines as the holding time increases for the same transmission aggressiveness  $\mathbf{r}$ .

### B. Objectives

**Unsaturated system.** When a CSMA-based algorithm can stabilize any feasible arrival rate  $\lambda \in C(G)$ , the algorithm is called *throughput optimal*. Intuitively, when  $s_i(\mathbf{r}^*) > \lambda_i$  for all link  $i$ , the arrival  $\lambda$  can be stabilized with transmission aggressiveness  $\mathbf{r}^*$ . A question to address is:

**QT:** For any  $\lambda \in C(G)$ , is there any transmission aggressiveness  $\mathbf{r}$  such that  $s_i(\mathbf{r}) \geq \lambda_i$  for all link  $i$ ? If there exists such  $\mathbf{r}$ , what are the CSMA algorithms that provide the transmission aggressiveness  $\mathbf{r}$  over long-term without any message passing and explicit knowledge of the given arrival rate  $\lambda$ ?

**Saturated system.** In this case, each link is assumed to be infinitely backlogged. Thus, CSMA algorithms are exploited to control the service rate of each link so as to make the long-term service rate close to some point of the boundary of  $C(G)$ , formally, a solution of the following optimization problem:

$$\max_{\gamma} \sum_{i \in \mathcal{L}} U(\gamma_i) \quad \text{subject to } \gamma \in C(G) \quad (2)$$

where  $U(\cdot)$  denotes an utility function with the nice properties such as concavity and differentiability. The questions to address in this case is:

**QU:** Let the solution of (2) be  $\gamma^*$ . How can we make each link have transmission aggressiveness to  $r_i^*$  so that  $s_i(\mathbf{r}^*) = \gamma_i^*$ ?

## III. CSMA: RESEARCH SURVEY

The research papers on optimal CSMA to date directly or indirectly address the questions **QT** and **QU**. In this section, we summarize them, starting the first two subsections by summarizing the results which can be arguably representative in terms of models and algorithms, followed by more extensions according to the criteria mentioned in Section I. Note that our presentation in terms of positioning and sequencing the papers cited here may be biased by the authors to some degree, and there may also be some missing references.

### A. Basic Results: Unsaturated

In [3], it is shown that, for any feasible arrival rate  $\lambda$ , there exists a finite transmission aggressiveness  $\mathbf{r}^*$  such that  $s_i(\mathbf{r}^*) \geq \lambda_i, \forall i \in N$ . From this, the authors conjectured that *throughput optimality* can be achieved by CSMA. We summarize the results on throughput-optimal CSMAs by classifying them into rate-based and queue-based approaches.

**Rate-based approach:** The authors in [3] propose a simple rate-based approach which allows transmission aggressiveness  $\mathbf{r}$  to converge to the  $\mathbf{r}^*$  with a time-scale assumption that the schedules from CSMA immediately follow a stationary distribution at each time slot. Later, Jiang et al. [4] shows that without the time-scale separation assumption, the proposed rate-based approach converges to  $\mathbf{r}^*$  for any strictly feasible arrival. The algorithm operates as follows:

*Step (1):* Each link  $i$  investigates packet arrival and schedule duration for a sufficient long time interval. Let link  $i$  adjust its transmission aggressiveness  $r_i(j)$  at time  $T(j)$  for  $j \in \mathbb{Z}^+$ .<sup>2</sup> Let  $\{A_i(t)\}$  and  $\{S_i(t)\}$  be arrival and scheduling process of link  $i$ , respectively. Then, the empirical arrival and service rates at  $T(j+1)$ , denoted by  $\hat{\lambda}_i(j)$  and  $\hat{s}_i(j)$ , respectively, are calculated by:

$$\begin{aligned} \hat{\lambda}_i(j) &= \frac{1}{T(j+1) - T(j)} \int_{T(j)}^{T(j+1)} A_i(t) dt \\ \hat{s}_i(j) &= \frac{1}{T(j+1) - T(j)} \int_{T(j)}^{T(j+1)} S_i(t) dt. \end{aligned}$$

*Step (2):* Link  $i$  adjusts its transmission aggressiveness  $r_i$  according to the empirical packet arrival and service rates as follows:

$$r_i(j+1) = r_i(j) + \beta(j)(\hat{\lambda}_i(j) - \hat{s}_i(j)), \quad (3)$$

where  $\beta(j)$  is a decreasing step size.

**Queue-based approach:** The rate-based approach is summarized as the scheme which directly estimates the demand and then provides the service rates to balance the demand and supply. A different approach can be developed by implicitly quantifying the supply-demand differential using a queue-length information, which we call queue-based approach. This queue-based CSMA can be regarded as an algorithm which emulates Max-Weight in a sluggish manner. By sluggish, we mean that the Markov chain induced by CSMA requires a time to reach a stationary distribution (close to what Max-Weight achieves).

In [11], the authors propose a scheme called Q-CSMA where  $r_i = f(Q_i)$ , where  $Q_i$  is the queue length of link  $i$  and  $f$  is a weight function. They prove that Q-CSMA is (throughput) optimal for any increasing function  $f$  under the time-scale separation assumption. Although they use a discrete time model, no collision exists due to synchronous operations (see Section III-D). Thus, the probability that a schedule is selected at each time slot, follows the stationary distribution (1). In other words, due to the choice of  $r_i = f(Q_i)$ , the probability to schedule  $\sigma$  is proportional to  $\exp(\sum_{i \in \mathcal{I}(G)} \sigma_i f(Q_i))$ , and it is negligible if the weight  $W(\sigma) = \sum_{i \in \mathcal{I}(G)} \sigma_i f(Q_i)$  is far from its maximum value (Max-Weight always chooses a schedule maximizing the weight).

The queue-based approach without time-scale separation has been first proposed and justified in [6] for special choices of weight function  $f$ , e.g.,  $f(x) = \log \log(x)$ . The key challenge in the work is to analyze a non-trivial correlation between queueing and scheduling dynamics (operating in the same time-scale) induced by a queue-based algorithm such as Q-CSMA. The authors in [6] resolve the correlation by (i) sufficiently slowing down the speed of the queueing dynamics

<sup>2</sup>We use  $j$  to index the state updates, and  $T(j)$  is the time of  $j$ -th update.

using a slowly increasing weight function  $f$ , such as  $f(x) = \log \log(x)$  and (ii) showing that scheduling dynamics run in a much faster time-scale than queueing dynamics in a certain sense. Due to some technical issues, we note that the CSMA in [6] requires a slight message passing to broadcast certain global information (e.g. the number of queues, the maximum queue-size) over the network. In the following work [8], the authors refine their approach toward removing the message passing. However, the maximum queue-size information still remains to be broadcasted, which was conjectured to be not necessary. The conjecture has been recently resolved in [7] using a certain distributed ‘learning’ mechanism: each node runs it to infer the maximum queue-size information without explicit message passing (and only using sensing information).

**Comparison.** The common goal of rate- and queue-based approaches is to control the CSMA parameters for the desired high performance, where they use the arrival rate or queue-size information for the control, respectively. The performance guarantees of rate-based algorithms are inherently sensitive to the assumption that the arrival rate is fixed (or very slowly changing), while queue-based ones are more robust against this issue, i.e., the queue-based results [6], [7], [8] hold even under time-varying arrival rates. However, analyzing queue-based algorithms are technically much harder, and hence the time-scale separation assumption or the information of the maximum queue length has been often used for technical convenience.

### B. Basic Results: Saturated

If each link has infinity backlog, the object of CSMA algorithms is to maximize network utility rather than stabilize the queues of links. In [5], utility optimality is considered for flows under the time-scale separation assumption. The algorithm in [5] considers a joint scheduling (via CSMA) and congestion control problem as follows:

$$\begin{aligned} \max_{\mu \in \Omega, \lambda \in [0,1]^n} & - \sum_{\sigma \in \mathcal{I}(G)} \mu_\sigma \log \mu_\sigma + V \left( \sum_{i \in \mathcal{L}} U_i(\lambda_i) \right) \\ \text{s.t.} & \mathbb{E}\{\sigma_i\} \geq \lambda_i, \quad \forall i \in \mathcal{L}, \end{aligned} \quad (4)$$

where  $V$  is some constant and  $\Omega$  is set of all probability measure on  $\mathcal{I}(G)$ . Then, the optimal solution turns out to be close to the utility optimal within  $\frac{\log |\mathcal{I}(G)|}{V}$  bound.

The formal proofs for saturated case without time-scale separation assumption are proposed in [9] and [4]. In [9], the authors provide an algorithm motivated by stochastic approximation controlled by Markov noise. At the starting time instance of each frame, similarly with (3), transmission aggressiveness is updated as follows: Each link  $i$  maintains its own virtual queue  $q_i$ , updated by:

$$q_i(j+1) = q_i(j) + \alpha(j) \left( U^{j-1} \left( \frac{q_i(j)}{V} \right) - \hat{s}_i(j) \right), \quad (5)$$

where  $V$  is some constant and  $\alpha(j)$  is a decreasing step size. Then, based on  $q_i(j)$ , CSMA runs with the backoff and holding times satisfying  $b_i(j+1)h_i(j+1) = \exp(q_i(j+1))$ . Similarly to (4),  $V$  controls the distance from optimality. The virtual queue length is a Lagrange multiplier that appears from the dual decomposition of the original objective (2), quantifying the demand-supply differential.

In [4], they also show that without time-scale separation, the optimal solution of the problem (4) can be achieved by primal-dual relationship as follows:

$$\begin{aligned} r_i(j+1) &= \max\{0, r_i(j) + \alpha(j)(\lambda_i(j) - \hat{s}_i(j))\} \\ \lambda_i(j+1) &= \arg \max_{y \in [0,1]} V \cdot U(y) - r_i(j+1)y. \end{aligned} \quad (6)$$

Note that the algorithms in [4] and [9] are essentially the same, because from the definition of  $r_i = \log(b_i \times h_i)$ , but there exists minor difference in their proof details.

The key rationale for the saturated case lies in the fact that the transmission aggressiveness is updated by quantifying the supply-demand differential, and the new aggressiveness is applied to the system with more modest updates with the belief that the system approaches to what is desired. The extension of utility optimal to multi-channel networks is provided in [10] without time-scale separation based on a much more simpler optimality proof. For faster convergence, a steepest coordinate ascent algorithm is proposed in [23]. Under this algorithm, at each time slot  $j$ , the transmission aggressiveness of link  $i$  is set to be proportional to the first derivative of utility function at empirical service rate, such that  $r_i = k \cdot U'(\gamma_i(j))$  where  $\gamma_i(j) = \frac{1}{j+1} \sum_{t=0}^j \hat{s}_i(t)$ .

### C. Time-scale Separation Assumption

In a Markov chain, it takes some time for a state to be close to a stationary regime. This time is called mixing time. In optimal CSMA algorithms, the transmission aggressiveness  $r(t)$ , which determines the transition rates (in continuous cases) and probabilities (in discrete cases), is time-varying. Thus, the main challenge in performance analysis of the optimal CSMA algorithms lies in the fact that such the mixing time can be much shorter than the change of transmission aggressiveness. In some papers, e.g., [3], [11], [13], [15], time-scale separation assumption, i.e., the assumption that a Markov chain can immediately reach a stationary distribution, has been made, which removes all the dirt in the proof.

As briefly mentioned in Sections III-A and III-B, two optimality proof techniques exist when no time-scale separation is assumed. First, the change of transmission aggressiveness is slowed down by taking a function of the parameter that affects the aggressiveness. For example, in [6], [7], [8], the queue length is such a parameter, where to represent the link weight,  $\log \log(Q_i)$  is used to make the regime that the speed of weight changes (thus, the speed of aggressiveness changes) becomes much slower than that of the mixing time. Another approach is to have an explicit device such as a step-size, which decreases with time. Examples include the work by [9] and [4] for the saturated case, where the step-size  $\alpha(j)$  plays such a role.

### D. Continuous/Discrete and Synchronous/Asynchronous

The assumption of continuous distributions of backoff and holding times, where most of work based on the continuous setting assumes exponential distributions, conveniently removes the need to consider collisions, leading to simple analysis. However, a real system is not continuous. For example, 802.11 operates based on the notion of a slot whose duration is 20  $\mu\text{sec}$ . In this discrete system, collisions naturally occur when two links content at a same slot. Then, a link  $i$ 's throughput becomes characterized in more complex way

by considering the transmission loss due to collisions. Note that in the discrete case, geometrically distributed backoff and holding times are used in the modeling because of its memoryless property.

Two directions are taken for discrete time systems in the papers. First, since the stationary distribution for the given backoff and holding times is decided by their product, not their individual values, the holding time can be arbitrarily large as long as the product is chosen as planned. This implies that the throughput loss by collisions can be sufficiently reduced by enlarging the holding times, so that their performance is almost close to what has been obtained in the continuous case. However, this may not be practical, because long holding times are very bad for short-term fairness. In [24], [9], the tradeoff between throughput and short-term fairness is asymptotically analyzed, where it is indeed required that a high cost of short-term fairness should be paid to increase throughput; where short-term fairness is defined as the inverse of the average delay between two successive successful transmissions. In [5], [7], for a desired transmission aggressiveness  $r_i$  for each link  $i$ , the authors propose throughput optimal algorithms with collisions, where the holding time of link  $i$  is proportional to  $\exp(r_i)$  with a fixed backoff time, so that the holding time consequently increases if a larger aggressiveness is needed. This approach shares the idea, mentioned earlier, that the enlarged holding time can reduce the throughput loss due to collisions. Second, as in [11], a synchronous system with frames, consisting of separate control and data phases, is designed so that, through slight message passing in the control phase, collisions is resolved.

When links operate under a common clock, the control actions can be time-synchronized, and thus, more efficient design is possible. Continuous systems, where continuity is assumed for theoretical purpose, is by nature asynchronous. More serious issues on synchronization are raised in discrete systems, for example, slots can be skewed, where guard time needs to be allocated, and loss of efficiency due to guard time overhead etc. requires more study. However, so far all discrete time based papers assume perfect synchronization.

#### E. Time-Varying vs. Fixed Channel

In modeling channels, most of the work assume that channel capacity is fixed. However, the channels are often time-varying in practice. Optimal CSMA over time-varying channels have recently investigated [16], [17]. In [16], CSMA under time-varying channels has been studied only for complete interference graphs, when the arbitrary backoff rate is allowed. The proof is based on the time-scale separation assumption, which does not often hold in practice and extremely simplifies the analysis (no mixing time related details are needed). In [17], the authors consider a channel model that the link capacity is randomly varied between 0 and 1 and the channel varying process is independent across links. Under this model, two canonical CSMA algorithms are considered: (i) A-CSMA which transmits a packet only if the capacity is 1 and (ii) U-CSMA which operates independently of the channel variation. Despite the intuition that A-CSMA may outperform U-CSMA due to its channel tracking ability, it is proved that U-CSMA can guarantee more throughput than A-CSMA, depending on the speed of channel variations, in particular, when the

speed of channel variation is fast. However, for slowly varying channel, A-CSMA achieves throughput optimality, whereas U-CSMA is suboptimal. Such performance difference comes from the mixing time of Markov chain, i.e., when the channels change faster than mixing time, A-CSMA may behave in an undesirable manner.

#### F. Delay

In addition to the “first-order” metric such as throughput or utility, the delay performance of optimal CSMA has been studied recently. Delay in optimal CSMA has been largely under-explored, where only a small set of work has been published with emphasis on the asymptotic results. Shah et al. [25] show that it is unlikely to expect a simple MAC protocol such as CSMA to have high throughput and low delay. Thus, to achieve  $O(1)$  delay, in [26], [27], modified CSMA algorithms are proposed. In [26], a modified CSMA requiring *coloring operation* achieves  $O(1)$  delay for networks with geometry (or polynomial growth). A *reshuffling* approach, which periodically reshuffles all on-going schedules under time synchronized CSMA, leads to both throughput-optimality and  $O(1)$  delay for torus (inference) topologies [27].

Without any modification, the algorithms that split the holding and backoff times for a desired transmission aggressiveness determine the delay. In this approach, mixing time has been a popular toolkit for delay analysis [26], [12]. Jiang et al. [12] proved that a discrete-time parallelized update algorithm achieves  $O(\log n)$  delay for a limited set of arrival rates. However, it was shown very recently [28] that mixing time based approach may not be the right way to capture delay dynamics even in the asymptotic sense. In [14], asymptotic variance is used for the other metric that measures delay. In this work, they arrange the CSMA algorithms by asymptotic variance and show that the algorithm reducing asymptotic variance enhances delay performance.

#### G. Toward Practice: Imperfect Sensing and Implementation

More practical situations start to be considered for optimal CSMA. First, in [13], the authors consider the case when sensing is imperfect. An example of imperfect sensing is the famous hidden terminal nodes. Other examples include false alarm (resp. miss detection), where a link can sense the idle (busy) medium as busy (idle) with a positive probability. False alarm is not highly critical to throughput optimality, but miss detection could reduce throughput since it generates collisions. In [13], the protocol, which overcomes miss detection, is proposed, which is provably throughput optimal, by letting each link operate with small backoff rate and long holding time.

In most of the aforementioned research, the physical layer is abstracted. For example, for interference model, the protocol model is used, assuming that packet transmission of a link depends on neighbor links only. In practice, success of a transmission is decided by whether its SINR is above a threshold or not, called SINR model. In [15], SINR model is considered with MIMO transmission. Under this model, each link can select a data rate and the transmission is successful when total interference is less than the marginal interference for the transmission rate. Even for the MIMO and SINR model, the authors propose an algorithm that achieve

throughput optimality with an assumption where each link has to have topological information.

A limited number of work on the implementation of optimal CSMA exists, mainly with focus on evaluation [18], [29]. They show that multiple adverse factors of practical occurrence not captured by the assumptions behind the theory can hinder the operation of optimal CSMA, introducing severe performance degradation in some cases [29]. In [20], [21], the interaction between TCP and optimal CSMA has been investigated due to the window based congestion control of TCP. Two algorithms each based on multiple sessions [20] or virtual queue mechanism [21], respectively was proposed. Very recently, a protocol, called O-DCF [22], reflecting the rationale of optimal CSMA, has been designed and implemented on the legacy 802.11 hardware, and shows significant performance improvement over the 802.11 DCF.

#### H. Open problems

In [7], [8], the throughput optimality of a queue-based CSMA algorithm is shown. However, to guarantee the throughput optimality, the choice of transmission aggressiveness  $r_i$  is not just  $r_i = f(Q_i)$  for some function  $f$ ; but  $r_i(t) = \max\{f(Q_i(t)), \sqrt{f(Q_{\max})}\}$  for  $f(x) = \log \log(x)$ ; where  $Q_{\max}$  denotes the maximum queue length over all links. Thus, every link has to know [8] or infer [7]  $Q_{\max}$  to calculate transmission aggressiveness. Showing throughput optimality of queue-base CSMA with  $r_i = f(Q_i)$  is still an open problem. In addition, it is not known yet whether the queue-based CSMA algorithm using other functions such as  $f(x) = \log(x)$  or  $f(x) = x$ , is throughput optimal or not.

Beyond the throughput optimality, designing a CSMA algorithm which is optimal in both throughput and delay remains quite open. In [25], the authors show that the task is impossible (unless  $RP=NP$ ) for arbitrary interference topology. However, the interference topology arising in practice is not arbitrary, but has certain geometric properties [27] or bounded degrees [12]. Hence, designing (and analyzing) CSMA algorithms toward delay optimality still remains a quite important promising challenge to be investigated in the future.

#### IV. CONCLUSION

An extensive array of analysis and protocols are proposed on what are efficient MAC schemes. Efficiency can be measured by control overhead, throughput, and fairness etc. This survey demonstrates that a simple, fully distributed MAC with no or little message passing, such as CSMA, can be designed to achieve optimality, where various findings have been explored, and people are starting to looking at their practical values by evaluation and implementation in real hardwares. Despite a long history of MAC research, there still exists under-explored areas toward simple, yet highly efficient MAC. We hope that this survey paper helps the readers with summarizing the current research progress on optimal CSMA.

#### REFERENCES

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

[2] Y. Yi and M. Chiang, *Next-Generation Internet Architectures and Protocols*. Cambridge University Press, 2011, chapter 9: Stochastic network utility maximization and wireless scheduling.

[3] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, June 2010.

[4] L. Jiang, D. Shah, J. Shin, and J. Walrand, "Distributed random access algorithm: Scheduling and congestion control," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6182–6207, dec. 2010.

[5] L. Jiang and J. Walrand, "Approaching throughput-optimality in distributed csma scheduling algorithms with collisions," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 816–829, June 2011.

[6] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *Proceedings of ACM Sigmetrics*, Seattle, WA, 2009.

[7] D. Shah, J. Shin, and P. Tetali, "Medium access using queues," in *Proceedings of Foundations of Computer Science (FOCS)*. IEEE, 2011, pp. 698–707.

[8] D. Shah and J. Shin, "Randomized scheduling algorithm for queueing networks," *Annals of Applied Probability*, vol. 22, pp. 128–171, 2012.

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

[10] A. Proutiere, Y. Yi, T. Lan, and M. Chiang, "Resource allocation over network dynamics without timescale separation," in *Proceedings of Infocom*, March 2010.

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

[12] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," in *Proceedings of Infocom*, 2011.

[13] T. H. Kim, J. Ni, R. Srikant, and N. Vaidya, "On the achievable throughput of csma under imperfect carrier sensing," in *Proceedings of Infocom*, april 2011, pp. 1674–1682.

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

[15] D. Qian, D. Zheng, J. Zhang, and N. Shroff, "CSMA-based distributed scheduling in multi-hop MIMO networks under SINR model," in *Proceedings of Infocom*, march 2010.

[16] B. Li and A. Eryilmaz, "A Fast-CSMA Algorithm for Deadline-Constrained Scheduling over Wireless Fading Channels," *ArXiv e-prints*, Mar. 2012.

[17] S.-Y. Yun, J. Shin, and Y. Yi, "Medium access over time-varying channels with limited sensing cost," *ArXiv e-prints*, 2012.

[18] J. Lee, J. Lee, Y. Yi, S. Chong, A. Proutiere, and M. Chiang, "Implementing utility-optimal CSMA," in *Proceedings of Allerton*, oct 2009.

[19] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. Knightly, and M. Chiang, "Experimental evaluation of optimal CSMA," in *Proceedings of Infocom*, april 2011.

[20] W. Chen, Y. Wang, M. Chen, and S. C. Liew, "On the performance of tcp over throughput-optimal CSMA," in *Proceedings of IWQoS*, june 2011.

[21] J. Lee, H.-W. Lee, Y. Yi, and S. Chong, "Improving TCP performance over optimal CSMA in wireless multi-hop networks," to appear in *IEEE Communication Letters*, 2012.

[22] J. Lee, Y. Yi, S. Chong, B. Nardelli, E. Knightly, and M. Chiang, "Making 802.11 DCF Optimal: Design, Implementation, and Evaluation," *ArXiv e-prints*, 2012.

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

[24] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Maximizing utility via random access without message passing," Microsoft Research Labs, UK, Tech. Rep., 2009.

[25] D. Shah, D. N. C. Tse, and J. N. Tsitsiklis, "Hardness of low delay network scheduling," *IEEE Transactions on Information Theory*, 2009, submitted.

[26] D. Shah and J. Shin, "Delay optimal queue-based CSMA," in *Proceedings of ACM Sigmetrics*, 2010.

[27] M. Lotfinezhad and P. Marbach, "Throughput-optimal random access with order-optimal delay," in *Proceedings of Infocom*, 2011.

[28] V. Subramanian and M. Alanyali, "Delay performance of CSMA in networks with bounded degree conflict graphs," in *Proceedings of ISIT*, aug 2011, pp. 2373–2377.

[29] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. Knightly, and M. Chiang, "Experimental evaluation of optimal CSMA," in *Proceedings of Infocom*, 2011.