# Distributed Medium Access for Camera Sensor Networks: Theory and Practice

Hojin Lee, Donggyu Yun, and Yung Yi

**Abstract** Camera sensor networks have recently emerged as an important class of sensor networks, where each node is equipped with a camera and has a capability of visually detecting events in its neighborhood. The applications of camera sensor networks are highly diverse, including surveillance, environmental monitoring, smart homes, and telepresence systems. In this article, we focus on one of the key unique characteristics of camera sensor networks: An event detected by a sensor node can trigger a large amount of sensing data generation, which should be delivered in a distributed manner, whereas in "traditional" sensor networks the volume of sensing data is typically small. Networking protocols to convey the captured image from sensors to decision making modules consist of from distributed and energy-efficient layers accessed via a high-throughput and low-delay MAC to fancy routing and transport protocols. In this article, we focus on the MAC layer and survey the theory and the practical implementation efforts of CSMA-based MAC mechanisms, referred to as optimal CSMA, that are fully distributed with the goal of guaranteeing throughput and delay.

**Key words:** Camera Sensor Network, Optimal CSMA, Throughput Optimization, Utility Optimization, Multi-channel

---

Hojin Lee
Electrical Engineering, KAIST, e-mail: `hojin.lee.79@gmail.com`

Donggyu Yun
Electrical Engineering, KAIST, e-mail: `dgyun@lanada.kaist.ac.kr`

Yung Yi
Electrical Engineering, KAIST, e-mail: `yiyung@kaist.edu`

# 1 Introduction

## 1.1 Camera Sensor Network

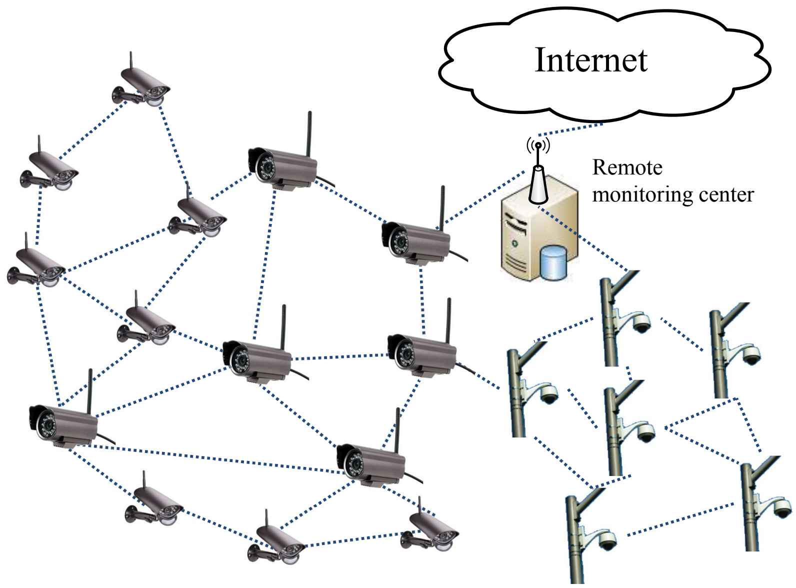



**Fig. 1** Camera Sensor Network

*Definition and Applications*

Camera sensor networks (CSN) are also called visual sensor networks, whose definition is presented in Wikipedia [36] as follows:

> A visual sensor network is a network of spatially distributed smart camera devices capable of processing and fusing images of a scene from a variety of viewpoints into some form more useful than the individual images.

Camera sensor networks can be applied to many types of useful applications, including:

- *Surveillance:* Surveillance has been the primary application of camera-based networks, where the monitoring of large public areas (such as airports, subways, etc.) is performed by a large number of security cameras. Cameras themselves usually produce just raw video streams. Thus, obtaining important and meaningful information from collected images necessitates a huge amount of local processing in the sensors as well as post-processing of them by delivering the images to the processing servers. This implies that both high-throughput wireless networks and smart processing engines are necessary to run CSNs efficiently.

- *Environmental monitoring:* Camera sensor networks can be used to monitor the areas that are remote and inaccessible, in which case energy-efficient operations, e.g., by duty cycling sensor nodes as in the conventional wireless sensor networks, to lengthen the lifetime of the networks. Traffics are generated on either event or time basis, depending on which the mechanism of operating the network should be different.
- *Telepresence:* Telepresence systems are the ones that enable remote users to virtually visit some location sensed by cameras. Examples include virtual museum or exhibition rooms equipped with live video cameras that are connected to the Internet and controlled by remote users. This case differs from the earlier two applications in that traffic patterns are "bi-directional" between sensors and users, although the traffic volume may be asymmetric.

*Example: CitySense [23]*

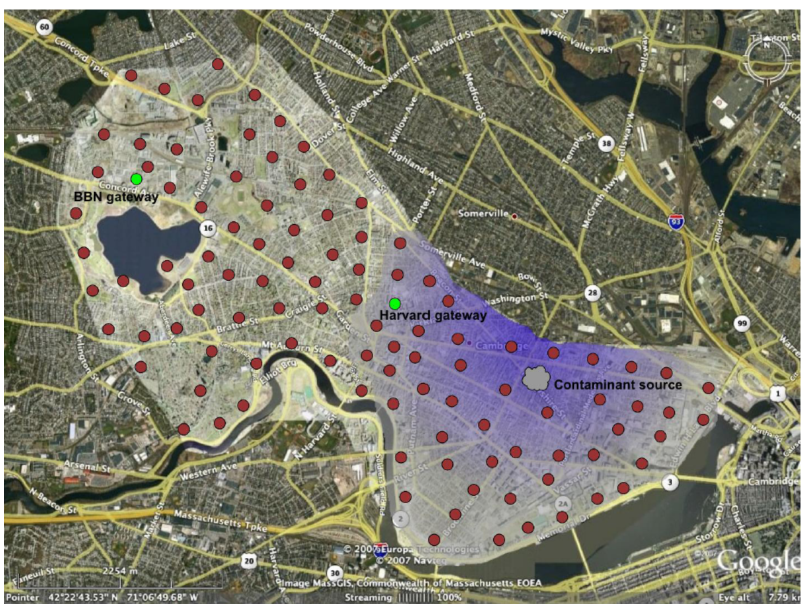



**Fig. 2** Node Deployment in Citysense. Source: [23].

As a nice example of camera sensor networks, we take *CitySense* project [23] that is an open, urban-scale wireless networking testbed with the goal of supporting the development and evaluation of novel wireless systems that span an entire city. CitySense consists of about 100 Linux-based embedded PCs outfitted with dual

802.11a/b/g radios and various sensors, mounted on buildings and streetlights across the city of Cambridge. The goal of CitySense is explicitly not to provide public Internet access, but rather to serve as a new kind of experimental apparatus for urban-scale distributed monitoring systems and networking research efforts.

*Networking and Data Delivery*

The key difference of camera sensor networks from other conventional sensor networks is the nature and the amount of information generated by each sensor. The captured visual data can be generated either periodically or on an event basis. In particular, sensor nodes capture a large amount of visual information which may be partially processed with the visual data from other cameras in the network, and thus changing the volume and the information from individual sensors. However, despite such in-network data processing, the volume of sensed data often still remains high, requiring high-performance wireless sensor networks. Also, it is often the case that the end-to-end data transmissions should satisfy low latency, thus requiring stable routing paths.

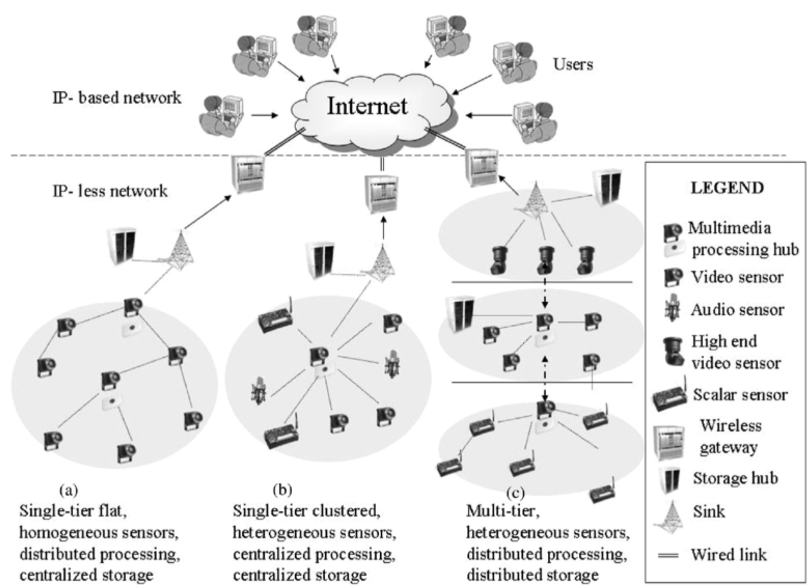



**Fig. 3** Reference Architecture of Camera Sensor Network. Source: [1]

Fig. 3 shows a reference architecture of CSN, proposed by [1], where a variety of connection types can be designed. Sensors can form a single-tier, flat or clustered network. A multi-tier architecture is also possible, where a group of sensors form one tier, connected to another tier through a gateway node. The network architecture can be selected differently, depending on different target applications, resource budget, and the scale size of the network.

## 1.2 Focus of This Chapter

Motivated by the fact that in CSNs a high volume of data is injected to the network by asynchronous events or periodic visual monitoring, and sensors should work in a fully distributed manner, we focus on how to deliver such large amount of traffic using a CSMA-based MAC, which is one of the famous, fully distributed MAC in the current practice. The popular 802.11 DCF, which can be a nice candidate MAC for CSNs, is a good example that based on CSMA. However, this chapter's focus is on providing the fundamental theories of running CSMA parameters, which guarantees a sense of optimal performance in terms of throughput and delay. These approaches have been extensively attempted in the name of *optimal CSMA*, as will be elaborated shortly. We note that in this chapter we do not explicitly consider energy-efficiency, but it can have high potential to be easily merged with optimal CSMA due to its fully distributed operation.

## 1.3 Optimal CSMA

### 1.3.1 Motivation

*CSMA (Carrier-Sense Multiple Access)*

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 feature 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.

*Wireless Scheduling: A Rough History*

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 [34] was published, in which 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., [37] 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 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).

### 1.3.2 Taxonomy

*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 the utility value of equilibrium rate is often the objective to be maximized. In terms of potential applications in camera sensor networks, unsaturated cases correspond to when sensing traffic is periodically generated, where periods can be deterministic or random, whereas event-driven visual sensors are well modeled by saturated cases, where when event occurs, a large volume of data traffic is generated so as to saturate the network temporarily.

*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.

*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.

*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.

*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.

*Theory vs. implementation*

Most of the work in the literature has 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 1. The rest of the paper is devoted to explaining their key concepts and brief summaries.

# 2 CSMA: A Theoretical Perspective

## *2.1 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 $G = (\mathscr{L}, E)$ denote the interference

**Table 1** Taxonomy of Optimal CSMA. TSS: Time-Scale Separation. This table is a extended version of that in [40].

| | Work | Sat/ Unsat | Cont/ /Disc | Sync/ Async | TSS | Summary and Comments |
|---|---|---|---|---|---|---|
| Theoretical Work | [7] | Unsat | Cont | Async | O | The first optimal CSMA with partial proofs |
| | [6] | Unsat | Cont | Async | × | More complete proof of [7] |
| | [8] | Unsat | Disc | Async | × | Throughput optimal with collision |
| | [28] | Unsat | Cont | Async | × | Queue based approach with full optimality proof without TSS |
| | [31] | Unsat | Disc | Async | × | Connecting Max-weight and CSMA with maximum queue size estimation |
| | [30] | Unsat | Cont | Async | × | Continuous time version of [31] |
| | [21] | Sat | Cont | Async | × | Utility optimal CSMA based on stochastic approximation with Markovian noise |
| | [26] | Sat | Cont | Async | × | Utility optimal CSMA under multiple channels |
| | [25] | Unsat | Disc | Sync | O | Queue based approach under synchronous system |
| | [5] | Unsat | Disc | Sync | × | Bounding delay based on parallel update of transmission aggressiveness |
| | [10] | Unsat | Disc | Async | O | Throughput optimal for imperfect carrier sensing |
| | [12] | Unsat | Cont | Async | × | Delay of optimal CSMA algorithms based on asymptotic variance |
| | [27] | Unsat | Cont | Async | O | MIMO and SINR-based interference model |
| | [18, 38] | Unsat | Cont | Async | × | CSMA over time-varying channel |
| | [13, 11] | Sat | Disc | Sync | × | Delay optimality of a throughput optimal CSMA |
| | [4] | Sat | Cont | Aync | × | Game-theoretic understanding of optimal CSMA |
| | [9, 39] | Sat | Cont | Aync | × | Approaching optimal CSMA with belief propagation in the theory of stochastic mechanics |
| | [19] | Unsat | Disc | Sync | × | Throughput optimal CSMA with worst-case delay guarantee |
| Impl. | [17, 24] | | Disc | Async | | Evaluation of optimal CSMA |
| | [2, 16] | | Disc | Async | | Study of interaction between CSMA and TCP |
| | [15] | | Disc | Async | | A new MAC and experimental validation on 802.11 hardware |

graph, where $\mathscr{L}$ and $E$ are the set of links and the set of edges between interfering links, respectively. We define by $\boldsymbol{\sigma} \triangleq [\sigma_i : i \in \mathscr{L}]$ [1] a scheduling vector for links in $G$. Since interfering links cannot successfully transmit a packet simultaneously, $\boldsymbol{\sigma}$ 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

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

where $n$ is the number of links. The feasible rate region (or capacity) $C = C(G)$ is convex hull of $\mathscr{I}(G)$, namely,

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

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 $\{\boldsymbol{\sigma}(t)\}$ of CSMA protocols becomes a time reversible Markov process. Then, the stationary distribution of a schedule $\boldsymbol{\sigma}$ is defined by $\boldsymbol{b} = [b_i]$ and $\boldsymbol{h} = [h_i]$:

$$\pi_{\boldsymbol{\sigma}}^{\boldsymbol{b},\boldsymbol{h}} = \frac{\prod_{i \in \mathscr{L}} (b_i h_i)^{\sigma_i}}{\sum_{\boldsymbol{\sigma}' \in \mathscr{I}(G)} \prod_{i \in \mathscr{L}} (b_i h_i)^{\sigma_i'}}, \tag{2}$$

which is a function of the product $b_i \times h_i$, for all $i \in \mathscr{L}$. Let $r_i = \log(b_i h_i)$ and $\boldsymbol{r} = [r_i]$, where $\boldsymbol{r}$ implicitly denotes transmission aggressiveness of links. From (2), the probability $s_i(\boldsymbol{r})$ that link $i$ is scheduled for $\boldsymbol{r}$, which is the link $i$'s throughput, is computed as follows:

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

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

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

slightly different from (2). However, the stationary distribution becomes close to (2) when the holding time $\boldsymbol{h}$ 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 $\boldsymbol{r}$.

### 2.2 Objectives

*Unsaturated system*

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

**(Q1)** For any $\boldsymbol{\lambda} \in C(G)$, is there any transmission aggressiveness $\boldsymbol{r}$ such that $s_i(\boldsymbol{r}) \geq \lambda_i$ for all link $i$? If there exists such $\boldsymbol{r}$, what are the CSMA algorithms that provide the transmission aggressiveness $\boldsymbol{r}$ over long-term without any message passing and explicit knowledge of the given arrival rate $\boldsymbol{\lambda}$?

*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:

$$\max_{\boldsymbol{\gamma}} \sum_{i \in \mathscr{L}} U(\gamma_i) \qquad \text{subject to} \quad \boldsymbol{\gamma} \in C(G) \tag{3}$$

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

**(Q2)** Let the solution of (3) be $\boldsymbol{\gamma}^*$. How can we make each link have transmission aggressiveness to $r_i^*$ so that $s_i(\boldsymbol{r}^*) = \gamma_i^*$?

## 3 Optimal CSMA: Survey

The research papers on optimal CSMA to date directly or indirectly address the questions **(Q1)** and **(Q2)**. 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 1. 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.

## *3.1 Basic Results: Unsaturated*

In [7], it is shown that, for any feasible arrival rate $\boldsymbol{\lambda}$, there exists a finite transmission aggressiveness $\boldsymbol{r}^*$ such that $s_i(\boldsymbol{r}^*) \geq \lambda_i,\ \forall i \in \mathscr{N}$. 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 [7] propose a simple rate-based approach which allows transmission aggressiveness $\boldsymbol{r}$ to converge to the $\boldsymbol{r}^*$ with a time-scale separation assumption that the schedules from CSMA immediately follow a stationary distribution at each time slot. Later, Jiang et al. [6] show that without the time-scale separation assumption, the proposed rate-based approach converges to $\boldsymbol{r}^*$ 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 $j \in \mathbb{Z}^+$. [2] 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:

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

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

$$r_i(j+1) = r_i(j) + \beta(j)(\hat{\lambda}_i(j) - \hat{s}_i(j)), \tag{4}$$

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 [25], 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 3.4). Thus, the probability that a schedule is selected at each time slot follows the stationary distribution (2). In other words, due to the choice of $r_i = f(Q_i)$, the probability to schedule $\boldsymbol{\sigma}$ is proportional to $\exp(\sum_{i \in \mathscr{N}(G)} \sigma_i f(Q_i))$, which becomes negligible if the weight $W(\boldsymbol{\sigma}) = \sum_{i \in \mathscr{N}(G)} \sigma_i f(Q_i)$ 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 [28] 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 [28] resolve the correlation by (i) sufficiently slowing down the speed of the queueing dynamics 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 [28] 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 [30], 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 [31] 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 [28, 31, 30] 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.

## 3.2 Basic Results: Saturated

If each link has infinite backlog, the object of CSMA algorithms is to maximize network utility rather than stabilize the queues of links. In [8], utility optimality is considered for flows under the time-scale separation assumption. The algorithm in [8] considers a joint scheduling (via CSMA) and congestion control problem as follows:

$$\begin{aligned} \max_{\mu \in \Omega, \boldsymbol{\lambda} \in [0,1]^n} \quad & \sum_{i \in \mathscr{L}} U_i(\lambda_i) - \frac{1}{V} \left( \sum_{\boldsymbol{\sigma} \in \mathscr{I}(G)} \mu_{\boldsymbol{\sigma}} \log \mu_{\boldsymbol{\sigma}} \right) \\ \text{s.t.} \quad & \mathbb{E}\{\sigma_i\} \geq \lambda_i, \ \ \forall i \in \mathscr{L}, \end{aligned} \tag{5}$$

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

The formal proofs for saturated case without time-scale separation assumption are proposed in [21] and [6]. In [21], the authors provide an algorithm motivated by stochastic approximation controlled by Markov noise.

> Time is divided into *frames* of fixed durations, $j = 1, 2, \cdots$. At the starting time instance of each frame, similarly with (4), transmission aggressiveness is updated as follows: Each link $i$ maintains its own virtual queue $q_i$, updated by:
>
> $$q_i(j+1) = q_i(j) + \alpha(j) \left( U'^{-1}\left( \frac{q_i(j)}{V} \right) - \hat{s}_i(j) \right), \tag{6}$$
>
> 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 (5), $V$ controls the distance from optimality. The virtual queue length is a Lagrange multiplier that appears from the dual decomposition of the original objective (3), quantifying the demand-supply differential.

In [6], they also show that without time-scale separation, the optimal solution of the problem (5) can be achieved by primal-dual relationship as follows:

$$
\begin{aligned}
r_i(j+1) &= \max\{0, r_i(j) + \alpha(j)(\lambda_i(j) - \hat{s}_i(j))\} \\
\lambda_i(j+1) &= \arg\max_{y \in [0,1]} V \cdot U(y) - r_i(j+1)y.
\end{aligned} \tag{7}
$$

Note that the algorithms in [6] and [21] are essentially the same, 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 to multi-channel networks is provided in [26] without time-scale separation based on a much more simpler optimality proof. For faster convergence, a steepest coordinate ascent algorithm is proposed in [3]. 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 $\gamma_i(j) = \frac{1}{j+1}\sum_{t=0}^{j} \hat{s}_i(t)$.

### *3.3 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 $\boldsymbol{r}(t)$, 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 the mixing time can be much longer than the change of transmission aggressiveness. In some papers, e.g., [7, 25, 10, 27], 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 3.1 and 3.2, 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 [28, 31, 30], 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 [21] and [6] for the saturated case, where the step-size $\alpha(j)$ plays such a role.

### *3.4 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 $\mu$sec. In this discrete system, collisions naturally occur when two links contend 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 [20, 21], 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 [8, 31], 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 [25], 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.

### *3.5 Channel: Time-Varying vs. Fixed*

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 been recently investigated [18, 38]. In [18], 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 [38], 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.

### *3.6 Imperfect Sensing and MIMO*

More practical situations start to be considered for optimal CSMA. First, in [10], 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 [10], 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 [27], 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.

# 4 Optimal CSMA: Multi-channel/Multi-radio

So far, we have discussed optimal CSMA for the basic setup, which is the single-channel/single-radio. However, to cope with a high volume of sensing traffic in camera sensor networks, the networks with more capacity may be necessary. A natural way of enlarging capacity is to build a network on top of multiple channels over multiple radios. This multi-channel/multi-radio system is not only important for widening the network capacity, but also for significantly reducing the delay. It has been reported that the naive optimal CSMA in general suffers from poor delay performance [22], because to achieve high throughput, once a CSMA schedule is determined, it needs to be frozen for a long time, i.e., high correlation of schedules. However, once channels are various, links can be "interleaved" appropriately so as to reduce correlation. In Section 4.1, we provide the model and the optimal algorithm for multi-channel/multi-radio systems, and then in Section 4.2, we will present that such multi-channel systems can significantly decrease delay, even achieving the order-wise delay optimality.

## *4.1 Optimal CSMA for Multi-channel/Multi-radio*

### 4.1.1 Model and Objective

*Network Model*

The network consists in a set $\mathscr{V}$ of $V$ nodes and a set $\mathscr{L}$ of $L$ links [3]. Denote by $s(l) \in \mathscr{V}$ and by $d(l) \in \mathscr{V}$ the transmitter and the receiver corresponding to link $l$. We also use the notation $v \in l$ if either $v = s(l)$ or $v = d(l)$. Node $v$ has $c_v$ radio interfaces or *radios* for short. On each link, data transmissions can be handled on any channel of a set $\mathscr{C}$ of $C$ channels. These channels are assumed to be orthogonal in the sense that two transmissions on different links and different channels do not interfere. We model interference by a symmetric boolean matrix $A \in \{0,1\}^{L\times L}$, where $A_{kl} = 1$ if link $k$ interferes link $l$ when transmitting on the same channel, and $A_{kl} = 0$ otherwise[4]. A node uses a radio interface to transmit or receive data on a given channel. Denote by $R_{cl}$ the rate at which $s(l)$ can send data to $d(l)$ on channel $c$.

[3] Note that the notations on the network model in this Section 4.1 slightly differ from those in other sections, e.g., in Section 2.1 and Section 4.2. For example, in Sections 2.1 and 4.2, we use $\mathscr{L}$ to refer to the set of nodes in the interference graph $G$, and $\mathscr{V}$ was not used there.

[4] The results can be readily extended to the case where the interference matrix may be different on different channels. In such case, interference would be modelled by $A \in \{0,1\}^{L\times L\times C}$ where $A_{klc} = 1$ iff link $k$ and $l$ interfere each other on channel $c$.

*Feasible Schedule Set and Feasible Rate Region*

Interference and the limited number of radios at each node impose some constraints on the set of possible simultaneous and successful transmissions on the various links and channels. We capture these constraints with the notion of schedule. A schedule $\boldsymbol{\sigma} \in \{0,1\}^{C \times L}$ represents the activities of the various links on the different channels: by definition, $\sigma_{cl} = 1$ if and only if link $l$ is active on channel $c$ (i.e., $s(l)$ is transmitting on channel $c$). A schedule $m$ is *feasible* if all involved transmissions are successful, i.e., if for all $k, l \in \mathscr{L}$ and all $v \in \mathscr{V}$,

$$(\sigma_{ck} = 1 = \sigma_{cl}) \Rightarrow (A_{kl} = 0) \qquad \text{(Interference constraint)}$$
$$\sum_{l \in \mathscr{L}: v \in l} \sum_{c \in \mathscr{C}} \sigma_{cl} \leq c_v \qquad \text{(Radio interface constraint)}$$

We define by $\mathscr{I}(G) \subset \{0,1\}^{C \times L}$ the set of the $M$ feasible schedules, which corresponds to the set of all feasible schedules in (1) for the single channel/single radio case.

We are now ready to define the *feasible rate region* $C = C(G)$ as the set of achievable long-term throughputs $\boldsymbol{s} = (s_l, l \in \mathscr{L})$ on the various links:

$$C(G) = \left\{ \boldsymbol{s} : \exists \boldsymbol{\alpha} \in [0,1]^M, \sum_{\boldsymbol{\sigma} \in \mathscr{I}} \pi_{\boldsymbol{\sigma}} = 1, \forall l \in \mathscr{L}, s_l \leq \sum_{\boldsymbol{\sigma} \in \mathscr{I}} \pi_{\boldsymbol{\sigma}} \sum_{c \in \mathscr{C}} \sigma_{cl} R_{cl} \right\}. \quad (8)$$

In the above expression, $\pi_{\boldsymbol{\sigma}}$ may be interpreted as the fraction of time schedule $m$ is activated.

*Objective: Saturated Case*

Naturally, we can study the optimal CSMA under multi-channel/multi-radio for both saturated and unsaturated cases, but in this section we focus only on the saturated case. As mentioned earlier, when the transmitters are saturated (i.e., they always have packets to send), the objective is to design a scheduling algorithm maximizing the network-wide utility, as formally given by:

$$\max \quad \Sigma_{l \in \mathscr{L}} U(\gamma_l), \quad \text{subject to} \quad \boldsymbol{\gamma} \in C. \qquad (9)$$

### 4.1.2 Optimal CSMA for Multi-channel/Multi-radio

*Multi-channel/Multi-radio CSMA with* $(\lambda_{cl}, b_{cl}, c \in \mathscr{C}, l \in \mathscr{L})$

The following extension of random back-off CSMA protocols can be considered for multi-channel/multi-radio systems. The transmitter of link $l$ has $C$ independent Poisson clocks, ticking at rates $\lambda_{cl}$, $c \in \mathscr{C}$. When a clock $c$ ticks, if the transmitter does have an available radio or if it is already transmitting or receiving on channel $c$, it does not do anything. Otherwise, it senses channel $c$, and checks whether the receiver has an available radio. If the channel is idle and if the receiver can receive data, it starts a transmission on channel $c$, and keeps the channel for an exponentially distributed period of time of average $b_{cl}$. Define $\lambda_{\cdot l} = (\lambda_{cl}, c \in \mathscr{C})$ and $b_{\cdot l} = (b_{cl}, c \in \mathscr{C})$, and denote by CSMA($\lambda_{\cdot l}, b_{\cdot l}$) the above access protocol. We also introduce $\boldsymbol{\lambda} = (\lambda_{\cdot l}, l \in \mathscr{L})$ and $\boldsymbol{b} = (b_{\cdot l}, l \in \mathscr{L})$. When each link $l$ runs CSMA($\lambda_{\cdot l}, b_{\cdot l}$), the network dynamics and performance can be analyzed using the theory of reversible Markov chains.

Let $\boldsymbol{\sigma}^{\boldsymbol{\lambda},\boldsymbol{b}}(t)$ be the active schedule at time $t$. Then $(\boldsymbol{\sigma}^{\boldsymbol{\lambda},\boldsymbol{b}}(t), t \geq 0)$ is a continuous-time reversible Markov chain whose stationary distribution $\boldsymbol{\pi}^{\boldsymbol{\lambda},\boldsymbol{b}}$ is given by

$$\forall \boldsymbol{\sigma} \in \mathscr{I}, \quad \pi_{\boldsymbol{\sigma}}^{\boldsymbol{\lambda},\boldsymbol{b}} = \frac{\prod_{l \in \mathscr{L}, c \in \mathscr{C}} (\lambda_{cl} b_{cl})^{\sigma_{cl}}}{\sum_{\boldsymbol{\eta} \in \mathscr{I}} \prod_{l \in \mathscr{L}, c \in \mathscr{C}} (\lambda_{cl} b_{cl})^{\eta_{cl}}},$$

where by convention $\prod_{l \in \emptyset}(\cdot) = 1$. Moreover, the link throughputs are given by

$$\forall l \in \mathscr{L}, \quad s_l^{\boldsymbol{\lambda},\boldsymbol{b}} = \sum_{\boldsymbol{\sigma} \in \mathscr{I}} \pi_{\boldsymbol{\sigma}}^{\boldsymbol{\lambda},\boldsymbol{b}} \sum_{c \in \mathscr{C}} \sigma_{cl} R_{cl}.$$

*Optimal Algorithm*

We now describe a generic algorithm that dynamically adapts these parameters so as to approximately solve the utility-maximization problem (9). Similarly to the optimal CSMA for the saturated case, time is divided into *frames* of fixed durations, $j = 1, 2, \cdots$, and the transmitters of each link update their CSMA parameters (i.e., $\lambda_{cl}, b_{cl}$) at the beginning of each frame. To do so, they maintain a virtual queue, denoted by $q_l(j)$ in frame $j$, for link $l$. The algorithm operates as follows:

1. During frame $j$, the transmitter of link $l$ runs CSMA($\lambda_{\cdot l}(j), b_{\cdot l}(j)$), and records the sum $\hat{s}_l(j)$ of the services received during this frame over all channels;
2. At the end of frame $j$, it updates its virtual queue according to

$$q_l(j+1) = \left[ q_l(j) + \alpha(j) \left( U'^{-1}\left( \frac{q_l(j)}{V} \right) - \hat{s}_l(j) \right) \right],$$

   and sets the $\lambda_{cl}(j+1)$'s and $b_{cl}(j+1)$'s such that their products are equal to $\exp\{R_{cl} q_l(j+1)\}$.

The above algorithm is highly similar to that for the single-channel/single-ratio, except that each transmitter of a link runs a multi-channel/multi-radio CSMA algorithm. Virtual queues are maintained per link, but per link/radio CSMA parameters are updated by those per link virtual queue length.

## *4.2 Delayed CSMA: Virtual Channel Approach*

### 4.2.1 Description for Delayed CSMA

The main idea of the *delayed* CSMA is to use multiple schedulers in a round-robin manner in order to effectively reduce the correlations between the link state process, in an attempt to alleviate the so-called *starvation problem*, i.e., once a schedule is chosen, it keeps being scheduled without any change for a large number of slots. Note that the algorithm and the setting in this section is for the case of single-channel/single-radio systems, which, however, shows that virtual multi-channel idea is able to reduce latency significantly. This gives a conjecture that physical multi-channel systems would have highly good delay performance. Different from the model in the time in the optimal CSMA for single- and multi-channel/radio systems, we take a *discrete* time-slotted model, indexed by $t = 1, 2, \cdots$ . for convenience. Delayed CSMA [11] is described as follows:

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

Here, $N_i = \{j \in \mathscr{L} : (i, j) \in E\}$ as the set of neighbors of link $i$. In the *delayed* CSMA, at each time slot, a decision schedule is chosen $\mathscr{D}(t) \in \mathscr{I}(G)$, which corresponds to a selection of an independent set of $G$. The active links in the decision schedule become the candidate links which may change their state. There are various ways to choose a decision schedule $\mathscr{D}(t) \in \mathscr{I}(G)$ at each time slot. For example, each link simply attempts to access the medium with a fixed access probability $a_i$ and then $i \in \mathscr{D}(t)$ with probability $a_i \prod_{j \in N_i}(1-a_j)$, or a randomized scheme with light control message exchanges can be used, as in [25]. In general, we assume that $\{\mathscr{D}(t)\}$ is a set of independent identical random variables such that $\Pr\{i \in \mathscr{D}(t)\} > 0$ for all $i$.

As we mentioned in Section 2.1, given the transmission aggressiveness $\boldsymbol{r} = [r_i]$, the schedule $\{\boldsymbol{\sigma}(t) : t \equiv k \pmod T\}$[5] forms a (discrete-time) irreducible and aperiodic Markov chain for $k = 0, 1, \dots, T-1$, e.g., the $k$-th Markov chain is $\{\boldsymbol{\sigma}(uT+k) : u = 0, 1, 2, \dots\}$. The common stationary distribution $\boldsymbol{\pi} = [\pi_{\boldsymbol{\sigma}}]$ is given by

$$\pi_{\boldsymbol{\sigma}} = \frac{1}{Z} \prod_{i \in \mathscr{L}} r_i^{\sigma_i}, \tag{10}$$

where $Z = \sum_{\boldsymbol{\sigma} \in \Omega} \prod_{i \in \mathscr{L}} r_i^{\sigma_i}$ is a normalizing constant. Hence, one can think that the algorithm utilizes multiple $T$ independent Markov chains (or schedulers). From their ergodicity, we know that for all $i \in \mathscr{L}$,

$$\lim_{t \to \infty} \frac{1}{t} \sum_{s=0}^{t-1} \sigma_i(s) = \Pr_{\pi}\{\sigma_i = 1\}.$$

There are several ways to find an appropriate transmission aggressiveness $[r_i]$ such that the long-term link throughput $\lim_{t \to \infty} \frac{1}{t} \sum_{s=0}^{t-1} \sigma_i(s)$ is greater than the arrival rate $\lambda_i$, as we mentioned in Section 3.

Thus, we assume that links initially start with the desired transmission aggressiveness here. Formally speaking, for given $\varepsilon$-admissible arrival rate $\boldsymbol{\lambda}$, we assume that

$$\lim_{t \to \infty} \frac{1}{t} \sum_{s=0}^{t-1} \sigma_i(s) = \Pr_{\pi}[\sigma_i = 1] \geq \lambda_i + \varepsilon, \text{ for all } i \in \mathscr{L}. \tag{11}$$

### 4.2.2 Delay-Optimality of Delayed CSMA

For $\boldsymbol{\lambda} \in C(G)$ and given $\varepsilon > 0$, we say that $\boldsymbol{\lambda}$ is $\varepsilon$-*admissible* if $\lambda_i + \varepsilon < \mu_i$, for all $i \in \mathscr{L}$ and some $\boldsymbol{\mu} = [\mu_i] \in C(G)$. When the arrival rate is $\varepsilon$-admissible, we can define the notion of delay-optimal scheduling algorithm as follows.

**Definition 1 (Delay-Optimality).** A scheduling algorithm is called *per-link delay-optimal* (or simply delay-optimal) [6], if for any $\varepsilon$-admissible arrival rate $\boldsymbol{\lambda}$ with $\varepsilon = \omega(1)$,

$$\limsup_{t \to \infty} \mathbb{E}\left[Q_i(t)\right] = O(1), \qquad \text{for all } i \in \mathscr{L},$$

where $Q_i(t)$ is the queue length of link $i$ at time $t$. In the above definition, the orders $\omega(1)$ and $O(1)$ are with respect to the network size $|\mathscr{L}|$, i.e., delay-optimality means that the *per-link* queue-size remains 'constant' as the network size grows.

To describe the analysis for the performance of *delayed* CSMA, we first introduce the necessary definitions of the *total variation distance* and the corresponding *mixing time* of the CSMA Markov chain. The total variation distance between two

[5] We say $t \equiv k \pmod T$ if $t - k$ is an integer multiple of $T$. It is called congruent modulo.

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

probability distributions $\boldsymbol{\eta} = [\eta_i]$ and $\boldsymbol{\nu} = [\nu_i]$ on state space $\Omega$ is

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

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

$$M^{(k)}(\delta) = \inf\left\{ s : \max_{\boldsymbol{\mu}(k)} \left\| \boldsymbol{\mu}(uT+k) - \boldsymbol{\pi} \right\|_{TV} \le \delta,\ \forall u \ge s \right\},$$

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

The following theorem states the delay-optimality of the delay-optimality of the *delayed* CSMA algorithm.

**Theorem 1.** *For any $\varepsilon$-admissible arrival rate $\boldsymbol{\lambda}$, there exists $T^* = O\left(\frac{1}{\varepsilon^3}\log M(\varepsilon/2)\right)$ such that for all $T > T^*$, the corresponding* delayed *CSMA algorithm is delay-optimal, more formally,*

$$\lim_{t\to\infty} \mathbb{E}[Q_i(t)] = O\left(\frac{1}{\varepsilon^4}\right), \qquad \text{for all } i \in \mathscr{L}.$$

The above theorem states that the per-link average queue-size is bounded by a constant for sufficiently large $T$, the number of independent CSMA schedulers. The purpose of choosing large $T$ is to effectively reduce the dependency among consecutive link states, which promotes much faster link state changes and hence alleviates the starvation problem. For the proof of the Theorem 1, refer to [13].

#### 4.2.3 Related Work on Delay Reduction

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. [32] 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 [29, 22], modified CSMA algorithms are proposed. In [29], 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 [22].

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 [29, 5]. Jiang et al. [5] 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 [33] that mixing time based approach may not be the right way to capture delay dynamics even in the asymptotic sense. In [12], 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.

## 5 Practical Protocol and Implementation

### *5.1 Research on Optimal CSMA Practice*

A limited number of work on the implementation of optimal CSMA exists, mainly with focus on evaluation [17, 24]. 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 [24]. In [2, 16], 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 [2] or virtual queue mechanism [16], respectively was proposed. Very recently, a protocol, called O-DCF [15], 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. Recently, an enhanced version of O-DCF, called A-DCF [14], was proposed to work better with TCP.

### *5.2 O-DCF*

This subsection describes O-DCF [15], which effectively bridges the gap between practice and theory in optimal CSMA. In O-DCF, a product of access probability (determined by contention window (CW) size in 802.11) and transmission length is set to be proportional to the supply-demand differential for long-term throughput fairness. A combination of access probability and transmission length is smartly taken, where an access probability is initially selected as a sigmoid function of queue length and searched by Binary Exponential Backoff (BEB) in a fully distributed manner to adapt to the contention levels in the neighborhood. Then, transmission length is suitably selected for long-term throughput fairness. The explanation of O-DCF is elaborated in the following.

### 5.2.1 System Architecture of O-DCF

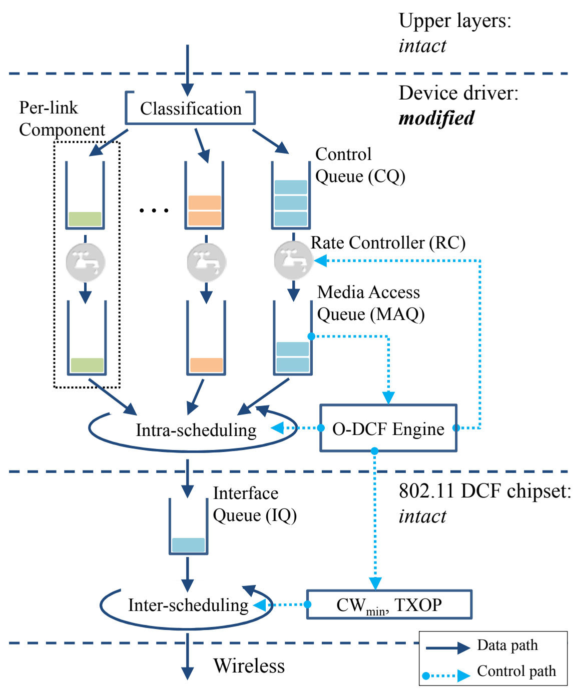



**Fig. 4** System Architecture of O-DCF.

In O-DCF, each node runs a per-neighbor control for accessing the medium by maintaining per-neighbor states, as shown in Fig. 4. Those states are used to determine how aggressively the node should access the medium in transmitting frames in a (link-level) destination-dependent manner. To this end, O-DCF maintains two per-neighbor queues: CQ (Control Queue) and MAQ (MAC Queue). CQ has the role of buffering the packets from upper layers, where each packet from upper layers is first classified according to its destination, and then enqueued into its per-neighbor CQ as frames. MAQ functions as a per-neighbor state that is importantly used to determine frames' medium access aggressiveness. A notion of Rate Controller (RC) resides between a CQ and a MAQ, and controls the dequeuing rate from the CQ to the MAQ. How the dequeuing rate is decided is critical in achieving fair medium access in O-DCF (see Section 5.2.2). Then, the service from a MAQ occurs when the HOL (Head-Of-Line) frame of the MAQ is moved into IQ (Interface Queue). 802.11 DCF parameters such as $CW_{min}$ and TXOP are appropriately set for controlling access aggressiveness. For multiple neighbors, the longest MAQ is served first; If the chosen transmission length exceeds a single frame size, multiple frames from the same MAQ are scheduled in succession.

### 5.2.2 Key Mechanisms of O-DCF

The MAQ maintains the supply-demand differential, and the dequeuing rate and the access aggressiveness are controlled by its queue length. For high performance, O-DCF translates the access aggressiveness into an adaptive combination of access probability and transmission length.

*Rate Control*

Let $Q_l(t)$ denote the length of MAQ for each link $l$ at time $t$. O-DCF controls the dequeuing rate from CQ to MAQ as follows:

$$\text{Rate from CQ to MAQ for link } l = \frac{V}{q_l(t)}, \tag{12}$$

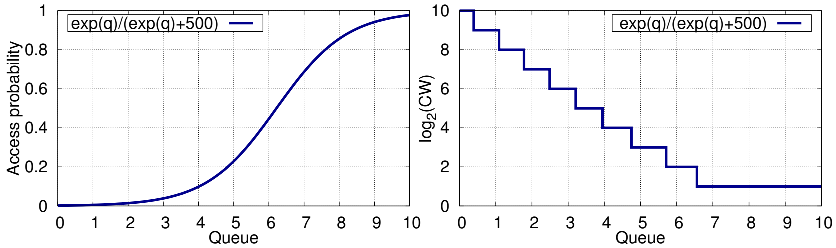
where $q_l(t) = bQ_l(t)$, and $b$ and $V$ are some constants. Intuitively, O-DCF decreases the rate for the long MAQ, and increases the rate when the MAQ is well-served. $b$ is a small value that corresponds to a step size, being responsible for slowing down the variations of queue length. $V$ is the constant that controls the sensitivity of dequeuing rate from CQ to MAQ. This form of dequeuing pattern is for achieving proportional fairness, derived from the log utility maximization; the dequeuing rate is $U'^{-1}(q_l(t)/V)$, where $U(\cdot)$ is a utility function, and $U(\cdot) = \log(\cdot)$ thus, $U'^{-1}(q_l(t)/V) = V/q_l(t)$. By suitably choosing the form of the utility function, various fairness criteria can be achieved.

*Access Aggressiveness Control*

CSMA has two critical parameters for controlling its aggressiveness: (i) access probability and (ii) transmission length. In many practical MACs such as 802.11, access probability is typically controlled by contention window (CW) size, and transmission length corresponds to the number of consecutive transmitted frames without separate media sensing. Aggressiveness simply means the product of access probability and transmission length, which are controlled differently for different neighboring links. Aggressiveness in O-DCF is basically controlled by the following simple rule:

$$\text{Aggressiveness (access prob.} \times \text{trans. length) for link } l = \exp(q_l(t)). \tag{13}$$

Intuitively, $q_l(t)$ tracks how well a link has been served over time. When a link has not been served for a long time, then it has high access aggressiveness by having either small CW size and/or long transmission length. How to choose the combination of CW size and transmission length is described next.




(a) Queue vs. $p$ (b) Queue vs. CW

**Fig. 5** Illustration of sigmoid function with respect to queue length.

*Adaptive Combination*

The key design aspects of O-DCF lies in which combination of access probability and transmission length should be chosen in practice to achieve high performance. When a frame (or a multiple of frames) from a MAQ is moved to IQ by the intra-scheduling for being ready for actual transmission, O-DCF's procedure of setting CSMA parameters is divided into the following three steps:

(1) *Initial access probability:* For a frame $f$ enqueued to IQ, using its per-neighbor state (i.e., its MAQ's length), an initial CW is smartly selected, where the basic principle is that the frames from under-served MAQs in terms of queue length are assigned smaller CWs. First, in order to effectively prioritize an under-served link, access probability of the link is calculated from a sigmoid function as shown in Fig. 5(a). Then, the access probability is converted into CW size conforming to the restriction of the 802.11 chipset [7] as in 5(b).

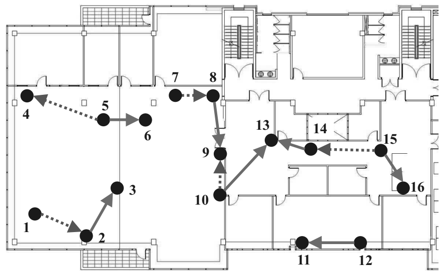
(2) *BEB for actual CW:* Once the initial CW size is chosen as a function of MAQ's length, the actual medium access is attempted, allowing BEB (Binary Exponential Backoff) to occur, which corresponds to a distributed search of the actual access probability.

(3) *Transmission length selection:* Once the actual CW is obtained after BEB, it is converted to an access probability, and then the transmission length is determined from (13) by considering the corresponding MAQ's length and the maximum transmission length specified in the legacy 802.11 chip.

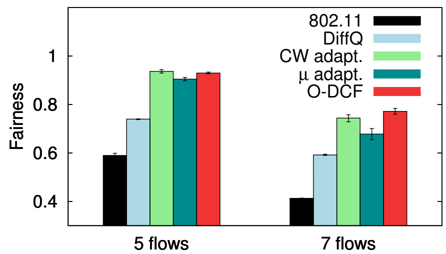
### 5.2.3 Performance Evaluation

O-DCF is compared with (i) 802.11 DCF, (ii) two versions of optimal CSMA in theory, and (iii) DiffQ [35]. For the standard optimal CSMA, two versions are tested to show the effect of the adaptive CSMA parameter combination in O-DCF: (i) *CW adaptation* in which the transmission length $\mu$ is fixed with a single packet

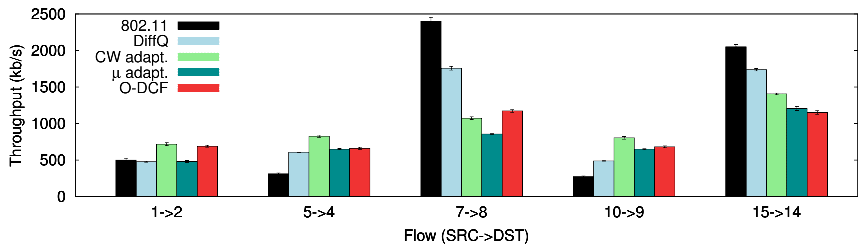
[7] CW sizes are one of values in $\{2^{i+1}-1 : i = 0,\ldots,9\}$.



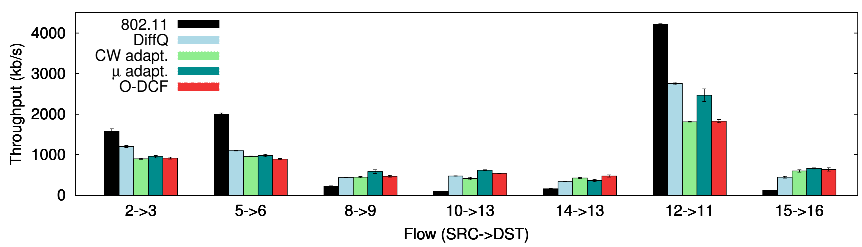

(a) Tested scenario




(b) Jain's fairness




(c) Per-flow throughput of 5 flows




(d) Per-flow throughput of 7 flows

**Fig. 6** Tested topology and performance comparison; (a) 16 nodes denoted by triangles are distributed in the area of 40m x 20m; dotted (solid) arrows represent 5 (7) flows for the first (second) scenario. (b) Jain's fairness comparison. (c)-(d) Per-flow throughput distributions.

and the access probability $p_l(t)$ is controlled, such that $p_l(t) \times \mu = \exp(q_l(t))$ [7], and (ii) $\mu$ *adaptation with BEB* (shortly, $\mu$ adaptation in this paper) in which the selection of $p_l(t)$ is delegated to 802.11 DCF and $\mu_l(t) = \exp(q_l(t))/p_l(t)$. Note that to understand the effect of different methods for the adaptation of CWs, $\mu$ adaptation is evaluated with BEB using 802.11's CW size, and is compare with O-DCF. DiffQ is a *heuristic* harnessing the 802.11e feature, and schedules the interfering links with different priorities based on queue lengths.

For performance comparisons, 16-node testbed is deployed as shown in Fig. 6(a). Each node is a netbook platform (1.66 GHz CPU and 1 GB RAM) running Linux kernel 2.6.31 and equipped with a single 802.11a/b/g NIC (Atheros chipset) running the modified MadWiFi driver for O-DCF's operations. To avoid external interference, a 5.805 GHz band in 802.11a is selected. The default link capacity is fixed with 6 Mb/s. In the 16-node testbed topology, two cases of five and seven concurrent flows under the default capacity are tested. This random topology enables to see how the algorithms perform in the mixture of hidden terminals and heavy contention scenarios including flow-in-the-middle (FIM) scenarios. The source and destination of each single-hop flow is chosen randomly. For each case, ten runs are repeated and error bars in all plots represent standard deviation. The duration of each run is 60 seconds.

Fig. 6(b) compares Jain's fairness achieved by all the algorithms for two scenarios. Over all the scenarios, O-DCF outperforms others in terms of fairness (up to 87.1% over 802.11 and 30.3% over DiffQ). The fairness gain can be manifested in the distribution of per-flow throughput, as shown in Fig. 6(c) and Fig. 6(d). O-DCF effectively prioritizes the flows with more contention degree (e.g., flow $10 \rightarrow 9$ forms *flow-in-the-middle* with flows $7 \rightarrow 8$ and $15 \rightarrow 14$) and provides enough transmission chances to highly interfered flows (i.e., $8 \rightarrow 9$, $10 \rightarrow 13$, and $14 \rightarrow 13$), compared with 802.11 DCF and DiffQ. The experimental topology is somewhat limited in size, tending to be full-connected. This leads to a small performance gap between the standard optimal CSMA and O-DCF, but 802.11 DCF yields severe throughput disparities of more than 40 times between flows $12 \rightarrow 11$ and $10 \rightarrow 13$ in the second scenario. Compared with 802.11, DiffQ performs fairly well in the sense that it prioritizes highly interfered flows. However, its access prioritization is heuristic, so there is still room for improvement compared with O-DCF.

## 6 Summary

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 hardware. Despite a long history of MAC research, there still exist 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.

**Acknowledgements** This work is supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as the Global Frontier Project.

## References


1. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: A survey on wireless multimedia sensor networks. Computer Networks **51**(4), 921–960 (2007)
2. Chen, W., Wang, Y., Chen, M., Liew, S.C.: On the performance of TCP over throughput-optimal CSMA. In: Proceedings of IWQoS (2011)
3. Hegde, N., Proutiere, A.: Simulation-based optimization algorithms with applications to dynamic spectrum access. In: Proceedings of CISS (2012)
4. Jang, H., Yun, S.Y., Shin, J., Yi, Y.: Distributed learning for utility maximization over CSMA-based wireless multihop networks. In: Proceedings of Infocom (2014)
5. Jiang, L., Leconte, M., Ni, J., Srikant, R., Walrand, J.: Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling. In: Proceedings of Infocom (2011)
6. Jiang, L., Shah, D., Shin, J., Walrand, J.: Distributed random access algorithm: Scheduling and congestion control. IEEE Transactions on Information Theory **56**(12), 6182 –6207 (2010)
7. Jiang, L., Walrand, J.: A distributed CSMA algorithm for throughput and utility maximization in wireless networks. IEEE/ACM Transactions on Networking **18**(3), 960 –972 (2010)
8. Jiang, L., Walrand, J.: Approaching throughput-optimality in distributed CSMA scheduling algorithms with collisions. IEEE/ACM Transactions on Networking **19**(3), 816–829 (2011)
9. Kai, C.H., Liew, S.C.: Applications of belief propagation in csma wireless networks. IEEE/ACM Transactions on Networking **20**(4), 1276–1289 (2012)
10. Kim, T.H., Ni, J., Srikant, R., Vaidya, N.: On the achievable throughput of csma under imperfect carrier sensing. In: Proceedings of Infocom (2011)
11. Kwak, J., Lee, C.H., Eun, D.Y.: A high-order markov chain based scheduling algorithm for low delay in csma networks. In: Proceedings of IEEE Infocom (2014)
12. Lee, C.H., Eun, D.Y., Yun, S.Y., Yi, Y.: From Glauber dynamics to Metropolis algorithm: Smaller delay in optimal CSMA. In: Proceedings of ISIT (2012)
13. Lee, D., Yun, D., Shin, J., Yi, Y., Yun, S.Y.: Provable per-link delay-optimal csma for general wireless network topology. In: Proceedings of IEEE Infocom (2014)
14. Lee, H., Moon, S., Yi, Y.: A-dcf: Design and implementation of delay and queue length based wireless mac. In: Proceedings of IEEE Infocom (2015)
15. Lee, J., Lee, H., Yi, Y., Chong, S., Nardelli, B., Chiang, M.: Making 802.11 DCF near-optimal: Design, implementation, and evaluation. In: Proceedings of IEEE SECON (2013)
16. Lee, J., Lee, H.W., Yi, Y., Chong, S.: Improving TCP performance over optimal CSMA in wireless multi-hop networks. IEEE Communication Letters **16**(9), 1388–1391 (2012)
17. Lee, J., Lee, J., Yi, Y., Chong, S., Proutiere, A., Chiang, M.: Implementing utility-optimal CSMA. In: Proceedings of Allerton (2009)
18. Li, B., Eryilmaz, A.: A Fast-CSMA Algorithm for Deadline-Constrained Scheduling over Wireless Fading Channels. ArXiv e-prints (2012)
19. Li, H., Vaidya, N.: Optimal CSMA-based wireless communication with worst-case delay and non-uniform sizes. In: Proceedings of IEEE Infocom (2014)
20. Liu, J., Yi, Y., Proutiere, A., Chiang, M., Poor, H.V.: Maximizing utility via random access without message passing. Tech. rep., Microsoft Research Labs, UK (2008)
21. Liu, J., Yi, Y., Proutiere, A., Chiang, M., Poor, H.V.: Towards utility-optimal random access without message passing. Wiley Journal of Wireless Communications and Mobile Computing **10**(1), 115–128 (2010)
22. Lotfinezhad, M., Marbach, P.: Throughput-optimal random access with order-optimal delay. In: Proceedings of Infocom (2011)
23. Murty, R.N., Mainland, G., Rose, I., Chowdhury, A.R., Gosain, A., Bers, J., , Welsh, M.: Citysense: An urban-scale wireless sensor network and testbed. In: Proceedings of IEEE HST (2008)
24. Nardelli, B., Lee, J., Lee, K., Yi, Y., Chong, S., Knightly, E., Chiang, M.: Experimental evaluation of optimal CSMA. In: Proceedings of Infocom (2011)
25. Ni, J., Tan, B., Srikant, R.: Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks. In: Proceedings of Infocom (2010)

26. Proutiere, A., Yi, Y., Lan, T., Chiang, M.: Resource allocation over network dynamics without timescale separation. In: Proceedings of Infocom (2010)
27. Qian, D., Zheng, D., Zhang, J., Shroff, N.: CSMA-based distributed scheduling in multi-hop MIMO networks under SINR model. In: Proceedings of Infocom (2010)
28. Rajagopalan, S., Shah, D., Shin, J.: Network adiabatic theorem: An efficient randomized protocol for contention resolution. In: Proceedings of ACM Sigmetrics. Seattle, WA (2009)
29. Shah, D., Shin, J.: Delay optimal queue-based CSMA. In: Proceedings of ACM Sigmetrics (2010)
30. Shah, D., Shin, J.: Randomized scheduling algorithm for queueing networks. Annals of Applied Probability **22**, 128–171 (2012)
31. Shah, D., Shin, J., Tetali, P.: Medium access using queues. In: Proceedings of IEEE FOCS (2011)
32. Shah, D., Tse, D.N.C., Tsitsiklis, J.N.: Hardness of low delay network scheduling. IEEE Transactions on Information Theory **57**(12), 7810–7817 (2011)
33. Subramanian, V., Alanyali, M.: Delay performance of CSMA in networks with bounded degree conflict graphs. In: Proceedings of ISIT (2011)
34. Tassiulas, L., Ephremides, A.: Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. IEEE Transactions on Automatic Control **37**(12), 1936–1949 (1992)
35. Warrier, A., Janakiraman, S., Ha, S., Rhee, I.: DiffQ: Practical differential backlog congestion control for wireless networks. In: Proceedings of IEEE Infocom (2009)
36. Wikipedia, V.S.N.: `http://en.wikipedia.org/wiki/Visual_sensor_network`
37. Yi, Y., Chiang, M.: Next-Generation Internet Architectures and Protocols. Cambridge University Press (2011). Chapter 16: Stochastic network utility maximization and wireless scheduling
38. Yun, S.Y., Shin, J., Yi, Y.: CSMA over time-varying channels: optimality, uniqueness and limited backoff rate. In: Proceedings of ACM Mobihoc (2013)
39. Yun, S.Y., Shin, J., Yi, Y.: Csma using the bethe approximation for utility maximization. In: Proceedings of ISIT (2013)
40. Yun, S.Y., Yi, Y., Shin, J., Eun, D.Y.: Optimal csma: A survey. In: Proceedings of ICCS (2012)