Computer Networks 122 (2017) 56–69

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A novel queue-length-based CSMA algorithm with improved delay characteristics

Dongyue Xue[a], Eylem Ekici[b,*], Rania Ibrahim[b,c], Moustafa Youssef[d]

[a] Department of Electrical and Computer Engineering, the Ohio State University, Columbus, OH, USA
[b] Computer Science Department, Purdue University, USA
[c] Alexandria University, Egypt
[d] Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology and Alexandria University, Alexandria, Egypt

ARTICLE INFO

Article history:
Received 12 September 2016
Revised 26 February 2017
Accepted 10 April 2017
Available online 19 April 2017

Keywords:
Queue length
CSMA
Delay characteristics

ABSTRACT

Recently, a group of queue-length-based CSMA algorithms have been proposed to achieve throughput optimality in wireless networks with single-hop transmissions. These algorithms suffer from two problems: (1) large delays, and (2) temporal starvation phenomenon, where communication links are inactive for a prolonged period of time before getting service. To mitigate these two problems, in this paper, we propose a novel $\mathbf{v}(t)$-regulated CSMA algorithm which can be implemented in a distributed manner using the RTS/CTS mechanism. Link scheduling is performed such that links with longer queues are favored so as to reduce average delay. The $\mathbf{v}(t)$-regulated CSMA algorithm also ensures a more frequent switch between schedules such that the effect of temporal starvation is reduced. The proposed algorithm is throughput optimal and achieves fully local implementation without global message passing. The thresholds to regulate the proposed algorithm are studied to optimize the upper-bound of the delay performance. We show through both hardware implementation and numerical evaluations that the algorithm indeed mitigates the temporal starvation problem and achieves far better delay performance than one of the other throughput-optimal CSMA algorithms.

1. Introduction

Efficient scheduling of wireless resources has always been one of the most challenging tasks for wireless networks. To achieve throughput-optimality, traditional back-pressure algorithms [1,2] calculate a maximal weight matching at each time slot. However, these algorithms need centralized scheduling with high complexity, and thus are not suitable for practical distributed implementations.

Recently, a class of distributed CSMA algorithms have been proposed in the literature [3–6] that achieve throughput optimality. In these algorithms, communication links are allowed to update their transmission based on a weight function of their queue lengths. We refer to these algorithms as regular throughput-optimal CSMA algorithms in the following discussion. We will use one of such algorithms as a benchmark in the implementation and analytical results later in this paper. Although these CSMA algorithms have been proved to be throughput-optimal [4,5], they suffer from the following problems: (1) *Temporal starvation*, defined in [10] as the phenomenon of links "being starved for prolonged periods indefinitely often despite having good stationary throughput". In other words, links usually undergo prolonged periods of inactivity followed by a prolonged period of activity. Temporal starvation leads to bursty service and undesirable jitter performance. The reason for this behavior is the operation of regular throughput-optimal CSMA algorithms: These algorithms schedule a link that was already active with high probability for prolonged periods, even if there are few (or even no) packets in its queue, during which its neighboring links suffer from starvation. (2) Undesirable delay performance [4,12]. This behavior of regular throughput-optimal CSMA algorithms also leads to the scheduling of links with short queues while there exist unscheduled links with longer queues in the network, resulting in long average packet delays.

There are a limited number of works analyzing the delay and temporal starvation problems in the literature. It has been shown in [9] that regular throughput-optimal CSMA algorithms achieve polynomial delay upper-bound for *a fraction* of capacity region in networks with single-hop transmissions. The effect of number of channels on temporal starvation is analyzed in [10]. Congestion control using virtual queues has been proposed in [12] in an at-

* Corresponding author.
E-mail addresses: xued@ece.osu.edu (D. Xue), ekici@ece.osu.edu (E. Ekici), ibrahimr@purdue.edu (R. Ibrahim), moustafa.youssef@ejust.edu.eg (M. Youssef).

http://dx.doi.org/10.1016/j.comnet.2017.04.036
1389-1286/

tempt to reduce delay without addressing the temporal starvation problem.

To address the delay and temporal starvation issues, in this paper, we propose a $\mathbf{v}(t)$-regulated CSMA algorithm that achieves fully local implementation without global message passing. Under the proposed algorithm, only links with weights above a certain threshold qualify to be scheduled. Since link weights are increasing functions of packet queue lengths, resources are scheduled to links with sufficiently large queue lengths only. This approach potentially alleviates the "persistent" scheduling problem of regular throughput-optimal CSMA algorithms. Compared to those algorithms, the $\mathbf{v}(t)$-regulated CSMA algorithm possesses the following two salient features: (1) Links with larger queue lengths are scheduled. By favoring longer queues over shorter ones, delays in longer queues are potentially reduced, and this reduction outweighs the increase in the delay of packets in the shorter, unserved queues. Thus, the average delay is potentially reduced. (2) The change in link schedules is more frequent. When an active link does not have sufficient data packets, the $\mathbf{v}(t)$-regulated CSMA algorithm requires the link to relinquish the wireless resource (i.e., channel). By handing over the channel much earlier, i.e., when the packet queue length drops below a threshold, the $v(t)$-regulated CSMA algorithm ensures a faster and more frequent switch between schedules, mitigating the temporal starvation problem.

While achieving an improvement on delay and temporal starvation, we prove that the $\mathbf{v}(t)$-regulated CSMA algorithm is *throughput-optimal*, as well. The proof is based on the time-scale separation assumption (i.e., the Markov chain of the schedules chosen by the scheduler is in steady state in each time slot) which has been employed in [3,4] and verified in [7,8]. Furthermore, our proposed CSMA algorithm is shown via both hardware implementation and simulations to have a much more favorable delay performance than a regular throughput-optimal CSMA algorithm [4] for the same set of arrival rate vectors. The temporal starvation problem is also shown to be mitigated significantly, where we use the second moment of inter-service intervals as the metric to characterize the degree of temporal starvation.

The rest of the paper is organized as follows: We propose the $\mathbf{v}(t)$-regulated CSMA algorithm and present its theoretical performance analysis in Section 2. Further discussions are provided in Section 3. Specifically, a method to approach time-scale separation is presented in Section 3.1, and we provide a guideline on choosing thresholds for the proposed algorithm in Section 3.2. We present the implementation results and numerical results in Sections 4 and 5, respectively. We conclude our work in Section 6.

2. v(t)-Regulated CSMA Algorithm

We introduce the network model in Section 2.1, with the proposal and theoretical performance analysis of the $\mathbf{v}(t)$-regulated CSMA algorithm presented in Sections 2.2 and 2.3, respectively. Before discussing details of the network model and algorithm design, we provide a summary of notations in Table 1.

2.1. Network model

Consider a wireless network with network topology $(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ denotes the node set and $\mathcal{L}$ denotes the set of single-hop directional communication links with $|\mathcal{L}| = L$. Each $l \in \mathcal{L}$ can be represented by $l = (m, n)$ as a single-hop flow from source m to destination n, for some $m, n \in \mathcal{N}$. We consider a general conflict graph interference model [3–6]. Specifically, for each link $l \in \mathcal{L}$, we define an interference set $\mathcal{N}_l \subseteq \mathcal{L}$, such that link l cannot transmit simultaneously with any link in $\mathcal{N}_l$. Without loss of generality, we let $l \in \mathcal{N}_l$, $\forall l \in \mathcal{L}$, and assume symmetric interference: $j \in \mathcal{N}_i$ if and only if $i \in \mathcal{N}_j$, $\forall i, j \in \mathcal{L}$. A set of links $\mathbf{x} \subseteq \mathcal{L}$ is called an independent set if none of the links in $\mathbf{x}$ interfere with each other, i.e., $i \notin \mathcal{N}_j$, $\forall i, j \in \mathbf{x}$ with $i \neq j$. We denote the set of all independent sets by $\mathcal{I}$ associated to the network topology $(\mathcal{N}, \mathcal{L})$.

We assume the considered wireless system is time-slotted, as is typical in many wireless standards (such as WLANs). We also assume that the transmission rate of each link is normalized and takes value in {0, 1}. Let $A_l(t)$ be the arrival process of the communication link $l \in \mathcal{L}$ over time slots t. For analytical simplicity, we assume the arrival process $A_l(t)$ is independent across $l \in \mathcal{L}$ and i.i.d. over time slots t with mean λ_l.[1] Without loss of generality, we assume that $A_l(t)$ is upper-bounded by some constant A_M, $\forall l \in \mathcal{L}$. At each time slot t, we represent a schedule by a vector $\mu(t) \triangleq (\mu_l(t))_{l\in\mathcal{L}}$, with $\mu_l(t) \in \{0, 1\}$ denoting the link rate schedule for link $l \in \mathcal{L}$. A schedule is said to be *feasible* if $\sum_{j\in\mathcal{N}_l\setminus\{l\}} \mu_j(t) = 0$, for any $l \in \mathcal{L}$ with $\mu_l(t) = 1$. Thus, when we associate each link $l \in \mathcal{L}$ with a packet queue length $Q_l(t)$, the corresponding queue dynamics can be written as:

$$Q_l(t+1) = [Q_l(t) - \mu_l(t)]^+ + A_l(t), \ \ \forall t \geq 0 \quad (1)$$

where the operator $[\cdot]^+ = \max\{0, \cdot\}$ and we assume that the arriving packets $A_l(t)$ are admitted to the packet queue at the end of each time slot t. The queue dynamics (1) can be equivalently represented as:

$$Q_l(t+1) = Q_l(t) - \mu_l(t) + A_l(t) + \beta_l(t), \quad (2)$$

where $\beta_l(t) \triangleq (\mu_l(t) - Q_l(t))\mathbf{1}_{\{Q_l(t)<\mu_l(t)\}}$ denotes the unused service for link l at time slot t, with $\mathbf{1}_{\{E\}}$ being an indicator function of the event E.

2.2. v(t)-regulated CSMA algorithm

Central to our proposed $\mathbf{v}(t)$-regulated CSMA algorithm is the establishment of a vector of thresholds $(\eta_l)_{l\in\mathcal{L}}$, such that, if the packet queue of an active link l has a link weight below threshold η_l, the active link l relinquishes the wireless resource and becomes idle. Since link weights are increasing functions of packet queue lengths, only links with *sufficiently large* queue lengths are scheduled. In comparison, under regular throughput-optimal CSMA algorithms, when a link occupies the channel, even if it has few packets (or even no packets) in its queue, it is highly likely that this link will remain scheduled for a considerably long period of time. By always scheduling links with sufficiently large queue lengths, the $\mathbf{v}(t)$-regulated CSMA algorithm potentially results in a reduction of packet delays in these scheduled queues, which outweigh the increase in delay of the packets in the other unscheduled queues (which have fewer packets). Hence, the $\mathbf{v}(t)$-regulated CSMA algorithm potentially reduces the average delay. In addition, under the proposed algorithm, the switch between schedules becomes more frequent than under regular throughput-optimal CSMA algorithms, mitigating the temporal starvation. Note that the distributed implementation of the algorithm is provided in Section 4.1.

In the following, we introduce definitions necessary for our CSMA algorithm. We first define the indicator variable $v_l(t) = \mathbf{1}_{\{w_l(t)>\eta_l\}}$, $l \in \mathcal{L}$, where $w_l(t)$ is the link weight of $l \in \mathcal{L}$ and η_l is the algorithm designed threshold for link l. We denote $\mathbf{v}(t) \triangleq \{v_l(t)\}_{l\in\mathcal{L}}$. Since we require that the $\mathbf{v}(t)$-regulated CSMA algorithm only schedule links l with link weights $w_l(t)$ larger than threshold η_l (i.e., $v_l(t) = 1$), we first define a $\mathbf{v}(t)$-regulated network topology $(\mathcal{N}, \mathcal{L}(\mathbf{v}(t)))$ generated based on the original topology $(\mathcal{N}, \mathcal{L})$. The $\mathbf{v}(t)$-regulated link set $\mathcal{L}(\mathbf{v}(t))$ is defined as:

$$\mathcal{L}(\mathbf{v}(t)) \triangleq \{l \in \mathcal{L} : v_l(t) = 1\},$$

[1] We note that the analysis can be readily extended to the case when $A_l(t)$ are Markovian over time.

Table 1
Summary of notations for the network model and the algorithm.

Notations	Definitions
$\mathcal{N}$	Node set of the network.
$\mathcal{L}$	Set of single-hop directional communication links with $\lvert\mathcal{L}\rvert = L$.
$l = (m, n)$	A single-hop flow from source m to destination n, for some $m, n \in \mathcal{N}$.
η_l	Algorithm designed threshold for link l.
$\mathbf{x}$	An independent set of links where none of the links in $\mathbf{x}$ interfere with each other.
$\mathcal{I}$	Set of all independent sets associated with the network topology $(\mathcal{N}, \mathcal{L})$.
$A_l(t)$	Arrival process of the communication link $l \in \mathcal{L}$ over time slots t.
$\mu_l(t)$	Link rate schedule for link $l \in \mathcal{L}$.
$Q_l(t)$	Packet queue length of link $l \in \mathcal{L}$.
$\beta_l(t)$	Unused service for link l at time slot t.
$w_l(t)$	Link weight of $l \in \mathcal{L}$ at time slot t.
$v_l(t)$	Indicator variable defined as $v_l(t) = \mathbf{1}_{\{w_l(t) > \eta_l\}}$.
$\mathcal{L}(\mathbf{v}(t))$	Set of links whose link weights $w_l(t)$ are greater than the thresholds η_l.

$\mathbf{v}(t)$-Regulated CSMA Algorithm:

1. Randomly select an independent set $\mathbf{x}(t) \in \mathcal{I}(\mathbf{v}(t))$ with probability (w.p.) $p_{\mathbf{x}(t)}$, such that:

$$\sum_{\mathbf{x}(t)\in\mathcal{I}(\mathbf{v}(t))} p_{\mathbf{x}(t)} = 1 \text{ and } \cup_{p_{\mathbf{x}(t)}>0}\, \mathbf{x}(t) = \mathcal{L}(\mathbf{v}(t)). \tag{3}$$

2. Scheduling link rates:

   2.1 $\forall l \in \mathbf{x}(t)$,

   2.1.1 If $\sum_{j\in\mathcal{N}_l\backslash\{l\}} \mu_j(t-1)v_j(t) = 0$:
   $\mu_l(t) = 1$, w.p. $p_l(t) \triangleq \frac{e^{w_l(t)}}{1+e^{w_l(t)}}$; $\mu_l(t) = 0$, w.p. $1 - p_l(t)$.

   2.1.2 Else, $\mu_l(t) = 0$.

   2.2 $\forall l \in \mathcal{L}\backslash\mathbf{x}(t)$, $\mu_l(t) = \mu_l(t-1)v_l(t)$.

Fig. 1. $\mathbf{v}(t)$-regulated CSMA algorithm - Link weights updated every T time slots.

i.e., $\mathcal{L}(\mathbf{v}(t))$ is the set of links whose link weights $w_l(t)$ are greater than the corresponding thresholds η_l. Under the $\mathbf{v}(t)$-regulated topology, we further define the set of all $\mathbf{v}(t)$-regulated independent sets $\mathcal{I}(\mathbf{v}(t))$ as:

$$\mathcal{I}(\mathbf{v}(t)) \triangleq \{\mathbf{x} \in \mathcal{I} : \forall l \in \mathbf{x}, v_l(t) = 1\} \subseteq \mathcal{I}.$$

In addition, we define $\mathbf{v}(t)$-regulated interference sets $\mathcal{N}_l(\mathbf{v}(t))$, $l \in \mathcal{L}$, as follows:

$$\mathcal{N}_l(\mathbf{v}(t)) \triangleq \{l \in \mathcal{N}_l : v_l(t) = 1\}, \forall l \in \mathcal{L}.$$

In Fig. 1, we propose the $\mathbf{v}(t)$-regulated CSMA algorithm. In Step 1, an independent set $\mathbf{x}(t)$ is selected probabilistically from $\mathcal{I}(\mathbf{v}(t))$. Links in $\mathbf{x}(t)$ are scheduled in Step 2.1, and other links are scheduled in Step 2.2. Specifically:

- In Step 2.1, for any link $l \in \mathbf{x}(t)$, if its neighboring links in the interference set $\mathcal{N}_l$ are not scheduled in the previous time slot *or* do not have a link weight above the threshold, i.e., $\sum_{j\in\mathcal{N}_l\backslash\{l\}} \mu_j(t-1)v_j(t) = 0$, then link l is scheduled service $(\mu_l(t) = 1)$ with link activation probability
$$p_l(t) \triangleq \frac{e^{w_l(t)}}{1+e^{w_l(t)}}. \tag{4}$$
Otherwise, $\mu_l(t) = 0$.
- In step 2.2, for any link l not belonging to $\mathbf{x}(t)$, $\mu_l(t) = 0$ when $v_l(t) = 0$; otherwise, the schedule for link l is unchanged, i.e., $\mu_l(t) = \mu_l(t-1)$.

From the selection of $\mathbf{x}(t)$ and Step 2.2, we know that $\mu_l(t) = 1$ only if $v_l(t) = 1$ (i.e., $w_l(t) > \eta_l$). The link weight $w_l(t)$ is defined as $w_l(t) = f(Q_l(t))$, where the link weight function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is chosen as follows: $f(x) = x$ in [3]; $f(x) = \log\log(x + e)$ in [5]; $f(x) = \log(x + 1)$ in [4,6]. It is easy to check that the above choices for function f satisfy the following properties:

- *Property (i):* f is an increasing function with $\lim_{x\to\infty} f(x) = \infty$.
- *Property (ii):* For any given $0 < \epsilon_3 < 1$, there exists $Q_M > 0$ such that $\forall x > Q_M$,
$$(1-\epsilon_3)f(x) < f(x-1) < f(x+A_M) < (1+\epsilon_3)f(x).$$
- *Property (iii):* $f'(x) \leq 1$, $\forall x \geq 0$.

Since $w_l(t)$ is an increasing function of queue length $Q_l(t)$, the $\mathbf{v}(t)$-regulated CSMA algorithm ensures that only links with sufficiently large queues (such that the corresponding link weights $w_l(t)$ are larger than the thresholds η_l) can be scheduled. Hence, an active link will switch to an idle state when it does not have a sufficiently large number of data packets in its queue, handing the resource over to other links with larger packet queues. On the other hand, under the regular throughput-optimal CSMA algorithms, even if an active link has few or no packets in its queue, it will continue to occupy the channel for prolonged periods with high probability, during which other links in its interference set suffers from starvation. Thus, with the $\mathbf{v}(t)$-regulated CSMA algorithm, the problem of temporal starvation is mitigated. Since service is scheduled only to links with sufficiently large queue lengths (i.e., links with packets having potentially large delays), the delay performance is also improved. The selection of the thresholds η_l is also essential to the algorithm performance, and a guideline on choosing the thresholds are provided in Section 3.2.

In Section 4.1, we introduce in detail a distributed implementation of the $\mathbf{v}(t)$-regulated CSMA algorithm, based on an RTS/CTS handshake mechanism. In the next subsection (Section 2.3), we show the throughput-optimality of the proposed algorithm. Implementation results and numerical analysis on delay performance and the temporal starvation issue are provided in Sections 4.2 and 5, respectively.

Table 2
Summary of propositions, lemmas, and theorems.

Statements	Purposes
Proposition 1 (Section 2.3)	Show feasibility of the schedule produced by the $\mathbf{v}(t)$-regulated CSMA algorithm.
Proposition 2 (Section 3.2)	Introduce a feasible choice of $(\eta_l)_{l\in\mathcal{L}}$ for the proposed $\mathbf{v}(t)$-regulated CSMA algorithm to minimize the delay upper bound.
Lemma 1 (Section 2.3)	Show schedules from $\mathbf{v}(t)$-regulated CSMA algorithm approximate a maximum weight matching scheduler with high probability.
Lemma 2 (Section 2.3)	Introduce an auxiliary stationary randomized algorithm to facilitate the proof of Theorem 1.
Lemma 3 (Section 2.4)	Show the $\mathbf{v}(t)$-regulated CSMA algorithm is equivalent to the scheduler in Fig. 2.
Lemma 4 (Appendix A)	Inequality for the proof of Theorem 2.
Lemma 5 (Appendix A)	Inequality for the proof of Theorem 2.
Theorem 1 (Section 2.3)	Show throughput optimality of the $\mathbf{v}(t)$-regulated CSMA algorithm.
Theorem 2 (Section 3.1)	Show throughput optimality of the modified $\mathbf{v}(t)$-regulated CSMA algorithm.

2.3. Throughput optimality of the $\mathbf{v}(t)$-regulated CSMA algorithm

In this subsection, we prove the throughput optimality of the proposed algorithm in Theorem 1. Before providing details of the proof, we enumerate the upcoming propositions, lemmas and theorems as well as their purposes in Table 2.

We will first show in Proposition 1 that the $\mathbf{v}(t)$-regulated CSMA algorithm always produces feasible schedules.

Proposition 1. *The schedule produced by the* $\mathbf{v}(t)$*-regulated CSMA algorithm is feasible, i.e., if* $\mathbf{y}(t-1)\in\mathcal{I}$*, then* $\mathbf{y}(t)\in\mathcal{I}$*, where* $\mathbf{y}(t)\triangleq\{l\in\mathcal{L}:\mu_l(t)=1\}$.

Proof. For any $l\in\mathbf{y}(t)$, we consider the following two cases:

- If $l\in\mathbf{y}(t)\cap\mathbf{x}(t)$, where $\mathbf{x}(t)$ is the independent set chosen according to Step 1 in Fig. 1, then $\mu_j(t-1)\nu_j(t)=0$, $\forall j\in\mathcal{N}_l\backslash\{l\}$, according to Step 2.1. Given any $j\in\mathcal{N}_l\backslash\{l\}$, we know that $j\notin\mathbf{x}(t)$, since $\mathbf{x}(t)$ is an independent set and $l\in\mathbf{x}(t)$. If $\nu_j(t)=0$, then $\mu_j(t)=0$ according to Step 2.2. Otherwise (i.e., when $\nu_j(t)=1$), $\mu_j(t-1)=0$, and hence $\mu_j(t)=\mu_j(t-1)=0$ according to Step 2.2. Therefore, $j\notin\mathbf{y}(t)$.
- If $l\in\mathbf{y}(t)\backslash\mathbf{x}(t)$, then $\mu_l(t-1)=1$ and $\nu_l(t)=1$ according to Step 2.2. For any given $j\in\mathcal{N}_l\backslash\{l\}$, we have

$$\sum_{k\in\mathcal{N}_j\backslash\{j\}}\mu_k(t-1)\nu_k(t)\geq\mu_l(t-1)\nu_l(t)=1. \tag{5}$$

  In addition, $\mu_j(t-1)=0$, since $\mu_l(t-1)=1$, i.e., $l\in\mathbf{y}(t-1)\in\mathcal{I}$.
  If $j\in\mathbf{x}(t)$, then $\mu_j(t)=0$ by (5) and Step 2.1.2. Otherwise (i.e., when $j\notin\mathbf{x}(t)$), from Step 2.2, $\mu_j(t)\leq\mu_j(t-1)=0$. Therefore, $j\notin\mathbf{y}(t)$.

Since the above analysis holds for any $l\in\mathbf{y}(t)$, we have shown that $\mathbf{y}(t)\in\mathcal{I}$, i.e., $\mathbf{y}(t)$ is an independent set: for any given $l\in\mathbf{y}(t)$, we have $j\notin\mathbf{y}(t)$, $\forall j\in\mathcal{N}_l\backslash\{l\}$. □

To support our analysis of the throughput performance, we introduce two related lemmas, Lemmas 1 and 2, to assist the proof of throughput optimality in Theorem 1. Specifically, in Lemma 1, we show that the schedules produced by the $\mathbf{v}(t)$-regulated CSMA algorithm approximate a maximum weight matching scheduler with high probability. In Lemma 2, we introduce an auxiliary stationary randomized algorithm.

Lemma 1. *Under* the time-scale separation assumption *(the Markov chain of the schedules chosen by the scheduler is in steady state in each time slot), for any given* ϵ_1 *and* δ_1 *satisfying* $0<\epsilon_1,\delta_1<1$*, we can find a constant* $B(\epsilon_1,\delta_1)>0$ *such that for any time slot t and with probability greater than* $(1-\delta_1)$*, the link rate scheduler finds a schedule* $(\mu_l(t))_{l\in\mathcal{L}}$*, satisfying:*

$$\sum_{l\in\mathcal{L}}w_l(t)\mu_l(t)\geq(1-\epsilon_1)\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}w_l(t),\ \text{ whenever } ||\mathbf{w}(t)||_\infty>B, \tag{6}$$

where $\mathbf{w}(t)\triangleq(w_l(t))_{l\in\mathcal{L}}$, $||\mathbf{w}(t)||_\infty\triangleq\max_{l\in\mathcal{L}}|w_l(t)|$*, and*

$$B\triangleq\max\left\{\frac{1}{\epsilon_1}\left(L\log 2+\log\frac{1}{\delta_1}\right),\ \frac{2}{\epsilon_1}\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}\eta_l\right\}.$$

For notational simplicity, we denote $||\cdot||\triangleq||\cdot||_\infty$ *in the following discussion. In (6),* $\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}w_l(t)$ *can be considered as the maximal weight matching over all feasible schedulers.*

Proof. The proof of Lemma 1 is provided in Section 2.4. □

We define the capacity region Λ as the set of all arrival rate vectors $(\lambda_l)_{l\in\mathcal{L}}$ supportable by the network, i.e., there exists a feasible scheduling algorithm, centralized or distributed, which is able to stabilize all the packet queues. Then, for any rate vector in Λ, there exists an (auxiliary) stationary randomized algorithm as stated in Lemma 2.

Lemma 2. *For any rate vector* $(\lambda_l)_{l\in\mathcal{L}}$ *strictly within the capacity region* Λ*, i.e., we can find some* $\epsilon_2>0$ *such that* $((1+\epsilon_2)\lambda_l)_{l\in\mathcal{L}}\in\Lambda$*, there exists a stationary randomized algorithm with schedules* $\left(\mu_l^{STAT}(t)\right)$ *independent of the queue lengths* $(Q_l(t))_{l\in\mathcal{L}}$*, such that, for any time slot t,*

$$\mathbb{E}\left\{\mu_l^{STAT}(t)\right\}=(1+\epsilon_2)\lambda_l,\forall l\in\mathcal{L}.$$

Similar formulations of randomized algorithm STAT and corresponding proofs have been given in [2,13], so we omit the proof of Lemma 2 for brevity.

A scheduling algorithm is throughput-optimal if the packet queues are stable in the mean [2,13] under the algorithm. The throughput optimality of the proposed algorithm is concluded in Theorem 1.

Theorem 1. *The* $\mathbf{v}(t)$*-regulated CSMA algorithm is* throughput-optimal, *for any arrival rate vector strictly within the capacity region* Λ*:*

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\{\left[\sum_{l\in\mathcal{L}}f^2(Q_l(t))\right]^{\frac{1}{2}}\right\}\leq\frac{B_2}{\epsilon}, \tag{7}$$

where $B_2>0$ *and* $0<\epsilon<1$ *are constants defined as follows:*

$$B_2\triangleq Lf(A_M)+Lf(Q_M)+Lf(Q_M+A_M)A_M+\gamma BL, \tag{8}$$

$$\epsilon\triangleq\min_{l\in\mathcal{L}}[(\gamma(1+\epsilon_2)-1)\lambda_l-\epsilon_3A_M]>0,$$

with $\gamma\triangleq(1-\epsilon_1)(1-\delta_1)$*. Note that* ϵ_1,δ_1,B *are defined in Lemma 1;* ϵ_2 *defined in Lemma 2; and* ϵ_3*,* Q_M *defined in Property (ii) of the link weight function f.*

Proof. The proof of Theorem 1 is given in Section 2.5. □

Since $\epsilon>0$ is required to ensure a positive upper-bound in (7), we must have from definition of ϵ that

$$\gamma>\frac{1}{1+\epsilon_2}, \tag{9}$$

An equivalent scheduler on $\mathbf{v}(t)$-regulated topology $(\mathcal{N}, \mathcal{L}(\mathbf{v}(t)))$:

1. Randomly select an independent set $\mathbf{x}(t) \in \mathcal{I}(\mathbf{v}(t))$ w.p. $p_{\mathbf{x}(t)}$, such that:

$$\sum_{\mathbf{x}(t)\in\mathcal{I}(\mathbf{v}(t))} p_{\mathbf{x}(t)} = 1 \text{ and } \cup_{p_{\mathbf{x}(t)}>0}\, \mathbf{x}(t) = \mathcal{L}(\mathbf{v}(t)).$$

2. Schedule link rates in $\mathcal{L}(\mathbf{v}(t))$:

   2.1 $\forall l \in \mathbf{x}(t)$,

   2.1.1 If $\sum_{j\in\mathcal{N}_l(\mathbf{v}(t))\backslash\{l\}} \mu_j(t-1) = 0$:

   $\mu_l(t) = 1$, w.p. $p_l(t)$; $\mu_l(t) = 0$, w.p. $1 - p_l(t)$.

   2.1.2 Else, $\mu_l(t) = 0$.

   2.2 $\forall l \in \mathcal{L}(\mathbf{v}(t))\backslash\mathbf{x}(t)$, $\mu_l(t) = \mu_l(t-1)$.

3. Schedule link rates in $\mathcal{L}\backslash\mathcal{L}(\mathbf{v}(t))$:

   $\mu_l(t) = 0$, $\forall l \in \mathcal{L}\backslash\mathcal{L}(\mathbf{v}(t))$. Or equivalently, $\mu_l(t) = 0$, $\forall l \in \mathcal{L}$ such that $w_l(t) \le \eta_l$.

Fig. 2. An equivalent scheduler for the proof in Lemma 1.

and

$$\epsilon_3 < \min_{l\in\mathcal{L}} \frac{[\gamma(1+\epsilon_2)-1]\lambda_l}{A_M}. \tag{10}$$

Since $\gamma < 1$ can be chosen arbitrarily close to 1 (due to the fact that ϵ_1 and δ_1 can be arbitrarily small according to their definitions in Lemma 1) and $\epsilon_3 > 0$ can be chosen arbitrarily small according to Property (ii) of the link weight function, there exist γ and ϵ_3 such that the inequalities (9) and (10) hold.

Note that the underlying Markov chain is positive recurrent due to the queue stability in the mean (7) according to Meyn and Tweedie [14], which implies the stability of the network [9].

2.4. Proof of Lemma 1

In this subsection, we prove Lemma 1 introduced in Section 2.3. We first show that the $\mathbf{v}(t)$-regulated CSMA algorithm is equivalent to the scheduler in Fig. 2 in the following lemma:

Lemma 3. *Given $\mu(t-1)$, $\mathbf{w}(t)$, and $\mathbf{v}(t)$, the $\mathbf{v}(t)$-regulated CSMA algorithm is equivalent to the scheduler in Fig. 2.*

Proof. Since Step 1 (of selecting an independent set $\mathbf{x}(t) \in \mathcal{L}(\mathbf{v}(t))$) is the same for both the $\mathbf{v}(t)$-regulated CSMA algorithm and the equivalent scheduler, proving Lemma 3 is equivalently to proving that the link schedules are equivalent under both algorithms given $\mathbf{x}(t)$. For any given $l \in \mathcal{L}$, we consider the following two cases:

- $l \in \mathbf{x}(t)$:
  Since
  $$\sum_{j\in\mathcal{N}_l\backslash\{l\}} \mu_j(t-1)v_j(t) = \sum_{j\in\mathcal{N}_l(\mathbf{v}(t))\backslash\{l\}} \mu_j(t-1),$$
  Step 2.1 under the $\mathbf{v}(t)$-regulated CSMA algorithm is equivalent to Step 2.1 under the equivalent scheduler. Thus, the schedule for $\mu_l(t)$, $l \in \mathbf{x}(t)$, is equivalent under both algorithms.
- $l \in \mathcal{L}\backslash\mathbf{x}(t)$:
  In this case, $\mu_l(t) = \mu_l(t-1)v_l(t)$ under the $\mathbf{v}(t)$-regulated CSMA algorithm. Since
  $$\mathcal{L}\backslash\{\mathbf{x}(t)\} = (\mathcal{L}(\mathbf{v}(t))\backslash\mathbf{x}(t)) \cup (\mathcal{L}\backslash\mathcal{L}(\mathbf{v}(t))),$$
  we consider the following two subcases under the equivalent scheduler. If $l \in \mathcal{L}(\mathbf{v}(t))\backslash\mathbf{x}(t)$, then according to Step 2.2 under the equivalent scheduler, $\mu_l(t) = \mu_l(t-1) = \mu_l(t-1)v_l(t)$. Otherwise (i.e., when $l \in \mathcal{L}\backslash\mathcal{L}(\mathbf{v}(t))$), according to Step 3 under the equivalent scheduler, $\mu_l(t) = 0 = \mu_l(t-1)v(t)$. Therefore, the scheduler for $\mu_l(t)$, $l \in \mathcal{L}\backslash\mathbf{x}(t)$, is equivalent under both algorithms.

Since the above discussion holds for any given $l \in \mathcal{L}$, we conclude that the two algorithms are equivalent. □

Step 1 and Step 2 of the equivalent scheduler in Fig. 2 form the regular throughput-optimal CSMA algorithm [9] with respect to the $\mathbf{v}(t)$-regulated topology $(\mathcal{N}, \mathcal{L}(\mathbf{v}(t)))$. The Proposition 2 in [4] proved the throughput optimality of the regular CSMA algorithm under the time-scale separation assumption. The proposition can be rephrased as follows: For any given $0 < \epsilon_1, \delta_1 < 1$, we can find $B_1(\epsilon_1, \delta_1) > 0$ such that, with probability greater than $(1-\delta_1)$, the equivalent scheduler (and hence the $\mathbf{v}(t)$-regulated CSMA algorithm according to Lemma 3) schedules $(\mu_l(t))_{l\in\mathcal{L}}$ satisfying the following:

$$\begin{aligned}
&\sum_{l\in\mathcal{L}} w_l(t)\mu_l(t) \\
&\ge (1-\frac{\epsilon_1}{2}) \max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(t))} \sum_{l\in\mathbf{x}} w_l(t), \text{ whenever } ||\mathbf{w}(t)|| > B_1, \\
&= (1-\frac{\epsilon_1}{2}) \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)v_l(t),
\end{aligned} \tag{11}$$

where $B_1 \triangleq \frac{1}{\epsilon_1}\left(L\log 2 + \log\frac{1}{\delta_1}\right)$.

When $||\mathbf{w}(t)|| > B$, we have:

$$\begin{aligned}
\frac{\epsilon_1}{2}\max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t) &\ge \frac{\epsilon_1}{2}||\mathbf{w}(t)|| > \frac{\epsilon_1 B}{2} \ge \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} \eta_l \\
&\ge \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)(1-v_l(t)),
\end{aligned}$$

where the last inequality follows the definition of $v_l(t)$. Hence, we obtain the following inequality, when $||\mathbf{w}(t)|| > B$,

$$\begin{aligned}
&\left(1-\frac{\epsilon_1}{2}\right)\max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t) \\
&< \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)(v_l(t)+1-v_l(t)) - \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)(1-v_l(t)) \\
&\le \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)v_l(t).
\end{aligned}$$

Combining (11), we know that, w.p. larger than $(1-\delta_1)$, whenever $||w(t)|| > B$,

$$\begin{aligned}
&\sum_{l\in\mathcal{L}} \mu_l(t)w_l(t) \\
&\ge \left(1-\frac{\epsilon_1}{2}\right)\max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)v_l(t)
\end{aligned}$$

$$> \left(1 - \frac{\epsilon_1}{2}\right)^2 \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t) \geq (1-\epsilon_1) \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t),$$

which concludes the proof of Lemma 1.

2.5. Proof of Theorem 1

In this subsection, we provide the proof for Theorem 1 introduced in Section 2.3. We first define the Lyapunov function $L(\mathbf{Q}(t)) \triangleq \sum_{l\in\mathcal{L}} g(Q_l(t))$, where $\mathbf{Q}(t) \triangleq (Q_l(t))_{l\in\mathcal{L}}$ and $g'(x) = f(x)$. We denote the corresponding Lyapunov drift as $\Delta(t) \triangleq \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\}$. From Taylor's Theorem, we have the following

$$\begin{aligned}\Delta(t) &= \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(\bar{Q}_l(t))(Q_l(t+1) - Q_l(t))|\mathbf{Q}(t)\right\} \\ &= \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(\bar{Q}_l(t))\beta_l(t)|\mathbf{Q}(t)\right\} \\ &\quad + \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(\bar{Q}_l(t))(A_l(t) - \mu_l(t))|\mathbf{Q}(t)\right\},\end{aligned}$$

where $\bar{Q}_l(t)$ lies between $Q_l(t)$ and $Q_l(t+1)$, $\forall l \in \mathcal{L}$.

Since $\beta_l(t) = 0$ if $Q_l(t) \geq 1$, we have $f(\bar{Q}_l(t))\beta_l(t) \leq f(Q_l(t+1))\mathbf{1}_{\{Q_l(t)=0\}} \leq f(A_M)$. Consequently,

$$\mathbb{E}\{\Delta(t)\} \leq Lf(A_M) + \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(\bar{Q}_l(t))(A(t) - \mu_l(t))\right\}. \tag{12}$$

To upper-bound the expectation of the Lyapunov drift $\mathbb{E}\{\Delta(t)\}$ in (12), we first find an upper-bound for $f(\bar{Q}_l(t))(A_l(t) - \mu_l(t))$. From Property (ii) of the link weight function f, for any given $\epsilon_3 > 0$, there exists $Q_M > 0$ such that $\forall Q_l(t) > Q_M$,

$$(1-\epsilon_3) f(Q_l(t)) < f(\bar{Q}_l(t)) < (1+\epsilon_3) f(Q_l(t)).$$

Utilizing the above property, if $Q_l(t) > Q_M$, we have

$$\begin{aligned}&f(\bar{Q}_l(t))(A_l(t) - \mu_l(t)) \\ &\quad < (1+\epsilon_3) f(Q_l(t))[A_l(t) - \mu_l(t)]^+ \\ &\qquad - (1-\epsilon_3) f(Q_l(t))[\mu_l(t) - A_l(t)]^+ \\ &\quad = f(Q_l(t))(A_l(t) - \mu_l(t)) + \epsilon_3 f(Q_l(t))|A_l(t) - \mu_l(t)| \\ &\quad \leq f(Q_l(t))(A_l(t) - \mu_l(t)) + \epsilon_3 A_M f(Q_l(t)).\end{aligned}$$

Hence, $f(\bar{Q}_l(t))(A_l(t) - \mu_l(t))$ can be bounded from above as follows:

$$\begin{aligned}&f(\bar{Q}_l(t))(A_l(t) - \mu_l(t)) \\ &\quad < f(Q_l(t))(A_l(t) - \mu_l(t))\mathbf{1}_{\{Q_l(t)>Q_M\}} \\ &\qquad + f(\bar{Q}_l(t))(A_l(t) - \mu_l(t))\mathbf{1}_{\{Q_l(t)\leq Q_M\}} + \epsilon_3 A_M f(Q_l(t)) \\ &\quad \leq f(Q_l(t))(A_l(t) - \mu_l(t)) + f(Q_l(t))(\mu_l(t) - A_l(t))\mathbf{1}_{\{Q_l(t)\leq Q_M\}} \\ &\qquad + \epsilon_3 A_M f(Q_l(t)) + f(Q_M + A_M)A_M \\ &\quad \leq f(Q_l(t))(A_l(t) - \mu_l(t)) + \epsilon_3 A_M f(Q_l(t)) \\ &\quad + f(Q_M) + f(Q_M + A_M)A_M,\end{aligned}$$

which leads to the following inequaility:

$$\begin{aligned}\mathbb{E}\{\Delta(t)\} &\leq Lf(A_M) + Lf(Q_M) + Lf(Q_M + A_M)A_M \\ &\quad + \sum_{l\in\mathcal{L}} \mathbb{E}\{f(Q_l(t))(\epsilon_3 A_M + A_l(t))\} \\ &\quad - \sum_{l\in\mathcal{L}} \mathbb{E}\{f(Q_l(t))\mu_l(t)\}.\end{aligned} \tag{13}$$

The last term of the RHS of the inequality (13) can be upper-bounded by

$$\begin{aligned}&-\sum_{l\in\mathcal{L}} \mathbb{E}\{f(Q_l(t))\mu_l(t)\} \\ &= -P(||\mathbf{w}(t)|| > B) \sum_{l\in\mathcal{L}} \mathbb{E}\{w_l(t)\mu_l(t)|(||\mathbf{w}(t)|| > B)\}\end{aligned} \tag{14}$$

$$\begin{aligned}&-P(||\mathbf{w}(t)|| \leq B) \sum_{l\in\mathcal{L}} \mathbb{E}\{w_l(t)\mu_l(t)|(||\mathbf{w}(t)|| \leq B)\} \\ &\quad \leq -\gamma P(||\mathbf{w}(t)|| > B)\mathbb{E}\{\max_{x\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t)|(||\mathbf{w}(t)|| > B)\} \\ &\quad \leq -\gamma P(||\mathbf{w}(t)|| > B) \sum_{l\in\mathcal{L}} \mathbb{E}\{w_l(t)\mu_l^{STAT}(t)|(||\mathbf{w}(t)|| > B)\} \\ &\quad = -\gamma \sum_{l\in\mathcal{L}} \mathbb{E}\{w_l(t)\mu_l^{STAT}(t)\} + \gamma P(||\mathbf{w}(t)|| \leq B) \\ &\qquad \times \sum_{l\in\mathcal{L}} \mathbb{E}\{w_l(t)\mu_l^{STAT}(t)|(||\mathbf{w}(t)|| \leq B)\} \\ &\quad \leq -\gamma \sum_{l\in\mathcal{L}} \mathbb{E}\{w_l(t)\mu_l^{STAT}(t)\} + \gamma BL,\end{aligned} \tag{15}$$

where we have employed Lemma 1 to (14) and substituted in (15) the stationary randomized algorithm STAT defined in Lemma 2.

Employing the above result to (13), we have

$$\begin{aligned}\mathbb{E}\{\Delta(t)\} &\leq B_2 + \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(Q_l(t))\left[\epsilon_3 A_M + A_l(t) - \gamma\mu_l^{STAT}(t)\right]\right\} \\ &= B_2 + \sum_{l\in\mathcal{L}} \mathbb{E}\{f(Q_l(t))[\epsilon_3 A_M + \lambda_l - \gamma\lambda_l(1+\epsilon_2)]\} \\ &\leq B_2 - \epsilon \sum_{l\in\mathcal{L}} \mathbb{E}\{f(Q_l(t))\},\end{aligned} \tag{16}$$

where we have employed Lemma 2 to (16).

Taking the time-average over $t = 0, 1, \ldots, T-1$ of both sides of (16) and taking the limsup with respect to T, we conclude

$$\begin{aligned}&\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{\left[\sum_{l\in\mathcal{L}} f^2(Q_l(t))\right]^{\frac{1}{2}}\right\} \\ &\leq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\sum_{l\in\mathcal{L}} f(Q_l(t))\} \leq \frac{B_2}{\epsilon},\end{aligned}$$

which proves (7).

3. Further discussions

In this section, we provide further discussions on the implementation issues of the $\mathbf{v}(t)$-regulated CSMA algorithm. Specifically, a modified $\mathbf{v}_T(t)$-regulated CSMA algorithm is proposed to approach time-scale separation in Section 3.1 and a feasible choice of thresholds is introduced in 3.2.

3.1. Approaching time-scale separation

We recall that Lemma 1 (and hence Theorem 1) in Section 2.3 is based on the time-scale separation assumption. This assumption requires the schedule determined by the algorithm to converge to its steady state faster than the rate at which link weights $w_l(t)$ change over time. In this section, we propose a method, referred to as $\mathbf{v}_T(t)$-regulated CSMA algorithm, to approximate this time-scale separation by updating the link weights less frequently.

The $\mathbf{v}_T(t)$-regulated CSMA algorithm is illustrated in Fig. 3. Specifically, we make the Markov chain of the schedules converge to the steady state distributions by updating the link weights

$\mathbf{v}_T(t)$-Regulated CSMA algorithm:

1. Randomly select an independent set $\mathbf{x}(t) \in \mathcal{I}(\mathbf{v}_T(t))$ w.p. $p_{\mathbf{x}(t)}$, such that:

$$\sum_{\mathbf{x}(t)\in\mathcal{I}(\mathbf{v}_T(t))} p_{\mathbf{x}(t)} = 1 \text{ and } \cup_{p_{\mathbf{x}(t)}>0}\, \mathbf{x}(t) = \mathcal{L}(\mathbf{v}_T(t)).$$

2. Scheduling link rates:

2.1 $\forall l \in \mathbf{x}(t)$,

2.1.1 If $\sum_{j\in\mathcal{N}_l\setminus\{l\}} \mu_j(t-1)v_{j,T}(t) = 0$:
$\mu_l(t) = 1$, w.p. $p_{l,T}(t) \triangleq \frac{e^{w_{l,T}(t)}}{1+e^{w_{l,T}(t)}}$; $\mu_l(t) = 0$, w.p. $1 - p_{l,T}(t)$.

2.1.2 Else, $\mu_l(t) = 0$.

2.2 $\forall l \in \mathcal{L}\backslash\mathbf{x}(t)$, $\mu_l(t) = \mu_l(t-1)v_{l,T}(t)$.

Fig. 3. $\mathbf{v}_T(t)$-regulated CSMA algorithm.

$w_{l,T}(t)$ in the $\mathbf{v}_T(t)$-regulated CSMA algorithm periodically every T time slots, $l \in \mathcal{L}$,

$$w_{l,T}(t) = f(Q_l(kT)),\ kT \le t < (k+1)T, \tag{17}$$

where T denotes the update period and k takes integer values. Similarly, we also update the indicator vector periodically as $\mathbf{v}_T(t) = (v_{l,T}(t))_{l\in\mathcal{L}}$ with

$$v_{l,T}(t) = v_l(kT),\ kT \le t < (k+1)T,\ l \in \mathcal{L}.$$

We note that the link activation probability is redefined as

$$p_{l,T}(t) \triangleq \frac{e^{w_{l,T}(t)}}{1+e^{w_{l,T}(t)}}.$$

The throughput-optimality still holds with the modified algorithm, which is formally stated in the following theorem.

Theorem 2. *The $\mathbf{v}_T(t)$-regulated CSMA algorithm is throughput-optimal.*

The proof of Theorem 2 is provided in Appendix A.

3.2. A guideline on choosing thresholds $(\eta_l)_{l\in\mathcal{L}}$

The selection of the thresholds $(\eta_l)_{l\in\mathcal{L}}$ is essential to the performance of the $\mathbf{v}(t)$-regulated algorithm. Since it is extremely hard to find closed-form results on packet delay for queue-length-based CSMA algorithms (though there are a few works in the literature that provide order results, e.g., [9]), instead of finding an optimal threshold that minimizes delay, we provide a guideline on the selection of thresholds $(\eta_l)_{l\in\mathcal{L}}$ in the following. Note that the RHS ($\frac{B_2}{\epsilon}$) of (7) can be considered as an upper-bound for packet queue lengths, which is an indicator on the delay performance. Thus, we choose the thresholds η_l, $l \in \mathcal{L}$, such that this upper-bound $\frac{B_2}{\epsilon}$ is minimized. In the following proposition, we introduce such a feasible choice of $(\eta_l)_{l\in\mathcal{L}}$ for the proposed $\mathbf{v}(t)$-regulated CSMA algorithm.

Proposition 2. *Given $\epsilon_1 = \delta_1 \le \frac{\epsilon_2}{2(1+\epsilon_2)}$,*

$$\eta_l = \eta_C \triangleq \frac{(L+1)\log 2 + \log\frac{1+\epsilon_2}{\epsilon_2}}{2\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{l\in\mathbf{x}}},\ l \in \mathcal{L}, \tag{18}$$

guarantees that $\frac{B_2}{\epsilon}$ is minimized.

Proof. According to the definition of B_2 in (8) and the definition of B in Lemma 1, it is sufficient to prove that the choice (18) ensures that

$$\frac{2}{\epsilon_1}\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}\eta_l \le B_1 = \frac{1}{\epsilon_1}\Big(L\log 2 + \log\frac{1}{\delta_1}\Big).$$

For any $l \in \mathcal{L}$, we obtain from (18) that

$$\frac{2}{\epsilon_1}\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}\eta_l = \frac{1}{\epsilon_1}\Big((L+1)\log 2 + \log\frac{1+\epsilon_2}{\epsilon_2}\Big) \le B_1,$$

completing the proof. Remark that since ϵ_1 and δ_1 can be chosen arbitrarily small, the given choice of ϵ_1 and δ_1 is feasible and ensures that the constraint (9) holds, i.e, $\gamma = (1-\delta_1)^2 > 1 - 2\delta_1 \ge \frac{1}{1+\epsilon_2}$. □

If local links do not have the knowledge of ϵ_2, which can be considered as the "distance" between the arrival rate vector and the maximal throughput, we can utilize a more conservative (smaller) choice of η_l:

$$\eta_l = \frac{(L+1)\log 2}{2\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{l\in\mathbf{x}}},\ l \in \mathcal{L}.$$

We note again that the value η_C does not necessarily minimize the delay or the expectation of packet queue lengths. Instead, we show in Proposition 2 that this choice is suboptimal in that it minimizes an upper-bound for queue lengths. Through numerical evaluations presented in Section 5, we show that choosing the threshold as η_C indeed leads to favorable delay performance and significantly mitigates the temporal starvation compared to a regular throughput-optimal CSMA algorithm.

4. Implementations

In this section, we first introduce a distributed implementation of the $\mathbf{v}(t)$-regulated CSMA algorithm in Section 4.1. We further implement the proposed algorithm in hardware on the Crossbow TelosB platform and present the implementation results in Section 4.2.

4.1. Distributed implementation of the v(t)-regulated CSMA algorithm

In the following, we present a distributed implementation of the $\mathbf{v}(t)$-regulated CSMA algorithm. This distributed implementation is based on the RTS/CTS mechanism. Note that the RTS/CTS handshake is only a tool to implement the distributed local interaction of the algorithm and other alternatives may exist. We assume each node has a single transceiver, i.e., a node cannot transmit and receive at the same time. For each node $n \in \mathcal{N}$, we define the following link set: $I_n(t) \triangleq \{(n,i) \in \mathcal{L} : v_{ni}(t) = 1\}$, where $v_{ni}(t)$ denotes $v_l(t)$ with $l = (n,i)$. For initialization, we let $\mu_l(0) = 0$, $\forall l \in \mathcal{L}$.

The main difficulty in the distributed implementation is to randomly select an independent link set satisfying condition ((3)). To achieve this, we employ the RTS/CTS mechanism. Specifically, in each time slot, we assign a number of T_s control mini-slots before

Distributed CSMA Algorithm at node n in time slot t

Step 1. Initialization:

A node r_n is selected in $I_n(t)$ uniformly at random;

n chooses a backoff time b_n uniformly at random from the first $(T_s - 1)$ mini-slots.

Step 2. RTS-CTS handshake:

for mini-slot $t_s = 1 : T_s$ (i.e., t_s is the current mini-slot)

  Step 2.1. The first micro-slot: CTS transmission

  if n has not overheard in the past any RTS(m), $\forall m \neq n$

   and n has not overheard any collision of RTS before

   and n has not sent out a CTS before

   and n has not received CTS(n) before

   and n received only one RTS(n) in mini-slot $(t_s - 1)$

   do: node n sends out a CTS responding to the RTS(n) received in mini-slot $(t_s - 1)$;

  end if

  Step 2.2. The second micro-slot: RTS transmission

  if n has not overheard in the past any CTS(m), $\forall m \neq n$

   and n has not overheard any collision of CTS before

   and n has not sent a CTS before **and** $b_n = t_s$

   do: node n sends out an RTS(r_n) to node r_n;

  end if

end for

Step 3. Schedule for links $\{(n,i) : (n,i) \in \mathcal{L}\}$:

$\mu_{ni}(t) = \mu_{ni}(t-1)v_{ni}(t), \forall i \neq r_n : (n,i) \in \mathcal{L}$.

if n received CTS(n) from r_n in Step 2 **do:**

  if $\sum_{j \in \mathcal{N}_{(n,r_n)} \setminus \{(n,r_n)\}} \mu_j(t-1)v_j(t) > 0$

   do: $\mu_{nr_n}(t) = 0$;

  else

  do: $\mu_{nr_n}(t) = 1$ w.p. $p_{(n,r_n)}$;

   $\mu_{nr_n}(t) = 0$ w.p. $(1 - p_{(n,r_n)})$;

  end if

else (i.e., (n, r_n) failed the RTS-CTS handshake)

  do: $\mu_{nr_n}(t) = \mu_{nr_n}(t-1)v_{nr_n}(t)$;

end if

Step 4. At the end of time slot t:

if $\mu_{nr_n}(t) = 1$ **do**:

  broadcast the updated $v_{(n,r_n)}(t+1)$ to neighboring nodes,

  i.e., nodes i such that there exists some node j with $(i,j) \in \mathcal{N}_{(n,r_n)}$.

end if

Fig. 4. Distributed implementation of the $\mathbf{v}(t)$-regulated CSMA algorithm.

data transmissions. The CSMA algorithm is executed at each node $n \in \mathcal{N}$ for each time slot t, as illustrated in Fig. 4. At the beginning of each time slot t, the node n selects a link (n, r_n) from $I_n(t)$ uniformly at random and chooses a random backoff time from the first (T_s-1) mini-slots (Step 1 in Fig. 4). Then under the conditions speficified in Step 2 in Fig. 4, node n initiates an RTS (*Request-To-Send*)-CTS (*Clear-To-Send*) handshake with its destination node r_n when the backoff duration is over. To facilitate the RTS/CTS handshake, each mini-slot is further split into two micro-slots (Step 2.1 and Step 2.2 in Fig. 4), dedicated for CTS and RTS transmissions, respectively. After the RTS/CTS handshakes, the CSMA algorithm assigns link rates (Step 3 in Fig. 4) following the $\mathbf{v}(t)$-regulated CSMA algorithm described in Section 2.2. At the end of each time slot t, the source node n of any scheduled link (n, r_n) is required to broadcast its updated $v_{(n,r_n)}(t+1)$ to its neighboring nodes (Step 4 in Fig. 4).

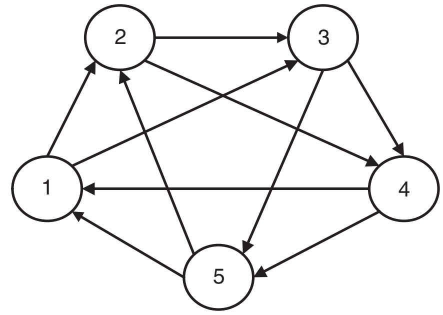

Fig. 5. 10-link network topology for implementation.

For narrative clarity, in Fig. 4 we let RTS(n) and CTS(n) denote, respectively, the RTS intended for node n and the CTS intended for node n, with $n \in \mathcal{N}$.

It is not difficult to check that under this distributed implementation, any link set $\mathbf{x}(t)$ whose links succeed in the RTS-CTS handshake (i.e., Step 2 in Fig. 4) is an independent link set and $\cup_{P(\mathbf{x}(t))>0}\mathbf{x}(t) = \mathcal{L}(\mathbf{v}(t))$, satisfying ((3)). Thus, the distributed algorithm is equivalent to the $\mathbf{v}(t)$-regulated CSMA algorithm introduced in Section 2.2.

4.2. Implementation results

In this section, $\mathbf{v}(t)$-regulated CSMA algorithm is validated in implementation vis-a-vis the QCSMA algorithm [4], a throughput-optimal queue-length-based algorithm. We employ the link weight function $f(x) = \log(x+1)$, since it is shown via simulation in [4,6] that the $\log(\cdot)$ form yields better delay performance than $f(x) = x$ [3] and $f(x) = \log\log(x+e)$ [5].

To quantify the degree of temporal starvation, we use the second moment of inter-service intervals as our metric. Specifically, $s_l(i)$ denoting the scheduled service time of ith packet of link $l \in \mathcal{L}$, we define the second moment of inter-service intervals J_l for link l as:

$$J_l \triangleq \lim_{I \to \infty} \frac{1}{I} \sum_{i=1}^{I} (s_l(i+1) - s_l(i))^2.$$

Note that a smaller J_l implies that link l is scheduled more frequently and its idle periods are shorter, indicating a lower level of temporal starvation. For analytical simplicity, in the implementation and simulation evaluation, we refer to the second moment of inter-service intervals as the average of J_l over all links $l \in \mathcal{L}$.

We implement the proposed algorithm (introduced in Section 4.1) and QCSMA algorithm in hardware on the Crossbow TelosB platform running NanoQplus OS [15]. Each Crossbow TelosB node is equipped with an IEEE 802.15.4 compliant RF transceiver and a programmable MSP430 processor [16]. To maintain the time-slotted structure, nodes periodically exchange timing information every 50 time slots to realign their internal clocks, which is sufficient in this experiment to achieve a data collision rate of less than 0.01%.

We employ a node-exclusive interference model (or one-hop interference model) under the 10-link topology in Fig. 5. We assume an identical arrival rate for all 10 communication links, i.e, $\lambda_l = \lambda$, $l \in \mathcal{L}$. Since at most two links can be active at each time slot un-

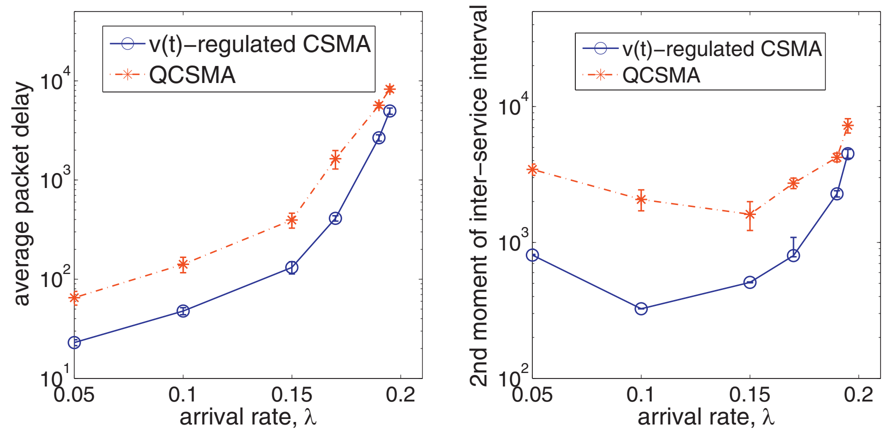

Fig. 6. Implementation results on delay and 2nd moment of inter-service intervals with respect to arrival rate under topology in Fig. 5, given $\eta_l = \eta_C,\ \forall l \in \mathcal{L}$.

der the node-exclusive model, the stabilizable range of λ is $0 < \lambda < 0.2$, with $\epsilon_2 = \frac{0.2}{\lambda} - 1$ characterizing the distance between λ and the maximum per-link throughput 0.2. The arrival processes $A_l(t)$ follow independent Bernoulli processes with parameter λ.

Fig. 6 illustrates the delay and the second moment of inter-service intervals under the two algorithms (where the time slot length is 360 ms with $T_s = 6$, and the confidence interval is 0.9), with $\eta_l = \eta_C,\ \forall l \in \mathcal{L}$. The implementation results are averaged over 5000 time slots to 90,000 time slots for different arrival rates in 5 runs such that the confidence interval is of acceptable range. When arrival rate is lower than 0.18 (90% of maximum throughput), the packet delay and the second moment of inter-service intervals, both averaged over all 10 links, are much smaller under the $\mathbf{v}(t)$-regulated CSMA algorithm than the QCSMA algorithm. The difference between the two algorithms in performance diminishes when the arrival rate further increases. This can be explained as follows: When the arrival rate increases towards the maximum (the value of 0.2 in this case), the queue lengths become larger, and hence the value of the threshold η_C ($\eta_C = 2.7$ in this implementation setup) becomes negligibly small compared to the queue length. Thus, the behavior of $\mathbf{v}(t)$-regulated CSMA algorithm approaches the QCSMA algorithm when the arrival rate λ is close to 0.2. However, even in a large-arrival-rate region, the $\mathbf{v}(t)$-regulated CSMA algorithm still yields much better performance. For instance, at an arrival rate $\lambda = 0.19$ (achieving 95% of maximum throughput), the packet delay is 53% smaller under the $\mathbf{v}(t)$-regulated CSMA algorithm than the QCSMA algorithm; and the second moment of inter-service intervals is 46% smaller.

We also note that under both algorithms, in a small-arrival-rate regime (e.g., $\lambda = 0.05$, achieving 25% of the maximum throughput), the inter-service intervals are potentially spaced farther apart because packet arrivals are farther apart, as well. Therefore, the second moment of inter-service intervals decreases when the arrival rate increases initially, as shown in Fig. 6. When arrival rate further increases, average queue lengths become larger, and it is more likely that an active link occupies the channel for prolonged periods of time followed by prolonged periods of inactivity, leading to a larger second moment of inter-service intervals, i.e., a higher level of temporal starvation.

5. Further numerical results

In this section, we further present a numerical comparative study of $\mathbf{v}(t)$-regulated CSMA algorithm with reference to the QCSMA algorithm. In the following two subsections, we illustrate the numerical performance using two different interference models under two different network topologies with the link weight function $f(x) = \log(x+1)$.

5.1. Numerical evaluation in a 10-link topology

In this subsection, we consider the same topology of Fig. 5 and the node-exclusive interference model with same arrival processes as in Section 4. To complement the implementation results, we further study the transient behavior of the system and the effect of thresholds $(\eta_l)_{l\in\mathcal{L}}$ on the algorithm performance.

When arrival rate $\lambda = 0.19$ (i.e., achieving 95% of the maximum stabilizable throughput), we find that all queues are stabilized and the throughput of 1.9 (summed over ten links) is indeed achieved. A snapshot of link rate schedules is shown in Fig. 7 with the suggested thresholds $\eta_l = \eta_C,\ \forall l \in \mathcal{L}$. To deliver a clear picture of instantaneous schedules, we only show the schedules for links (1, 2), (1, 3), (4, 1), and (5, 1) in Fig. 7. Compared to QCSMA algorithm under which a single link can occupy the channel over hundreds of time slots, the switch of link schedules is much more frequent under the $\mathbf{v}(t)$-regulated CSMA algorithm. Thus, the temporal starvation issue is successfully mitigated under the proposed algorithm.

We now study the effect of thresholds $(\eta_l)_{l\in\mathcal{L}}$ on the performance of the $\mathbf{v}(t)$-regulated CSMA algorithm. In Fig. 8 with arrival rate $\lambda = 0.1$, we show the performance of delay and second moment of inter-service intervals by varying the value of η, where we let $\eta_l = \eta,\ l \in \mathcal{L}$. The starred point in Fig. 8 denotes the case with $\eta_l = \eta_C,\ l \in \mathcal{L}$. We observe that η_C is indeed a favorable choice for the thresholds $(\eta_l)_{l\in\mathcal{L}}$ in that it leads to comparably good delay performance among all the thresholds. J_l performance could further be improved when η grows larger than η_C. However, we note that the decrease in J_l is trivial compared to the increase in delay as η becomes larger.

From Fig. 8, we see that even a small value of threshold (e.g., $\eta = 1$) can significantly reduce the delay and mitigate the issue of temporal starvation compared to the QCSMA algorithm. However, we notice that, as η gets larger, the delay increases. Recall that under $\mathbf{v}(t)$-regulated CSMA algorithm, only link weights (increasing functions f of queue lengths) greater than the threshold η can be scheduled. Thus, the average queue packet length is greater than or equal to $f^{-1}(\lfloor\eta\rfloor)$, where $\lfloor\cdot\rfloor$ denotes the floor function. When η grows significantly large, $f^{-1}(\lfloor\eta\rfloor)$ dominates the queue length. Thus, according to the Little's Law, the average packet delay increases accordingly in a large-η-regime.

We also observe that, when η increases, the duration of a link occupying the channel is expected to become smaller, leading to a more frequent change in schedules. Hence, the second moment of inter-service intervals J_l becomes smaller in Fig. 8. However, we

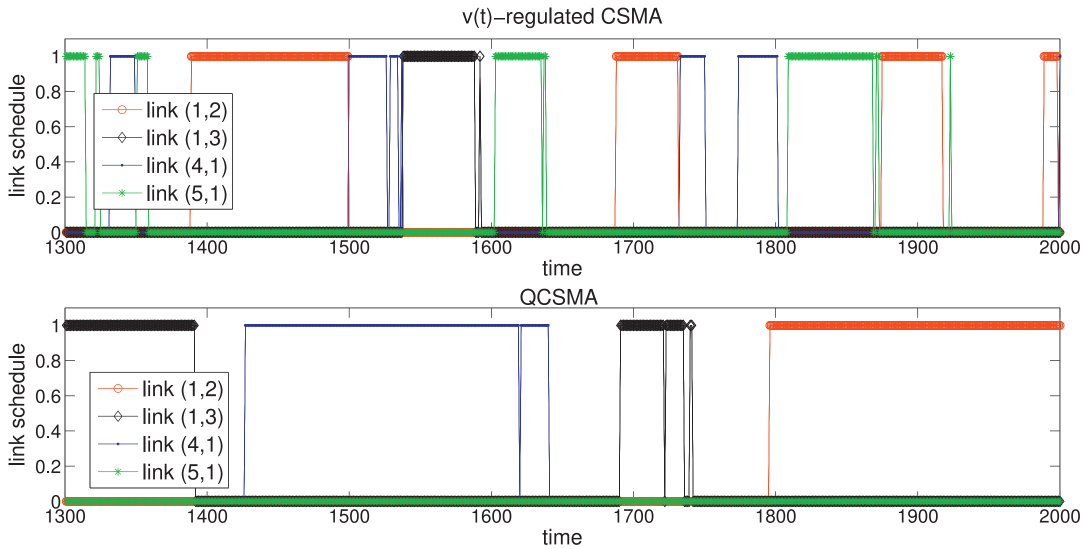

Fig. 7. Numerical results: Snapshot of Link schedules when arrival rate $\lambda = 0.19$.

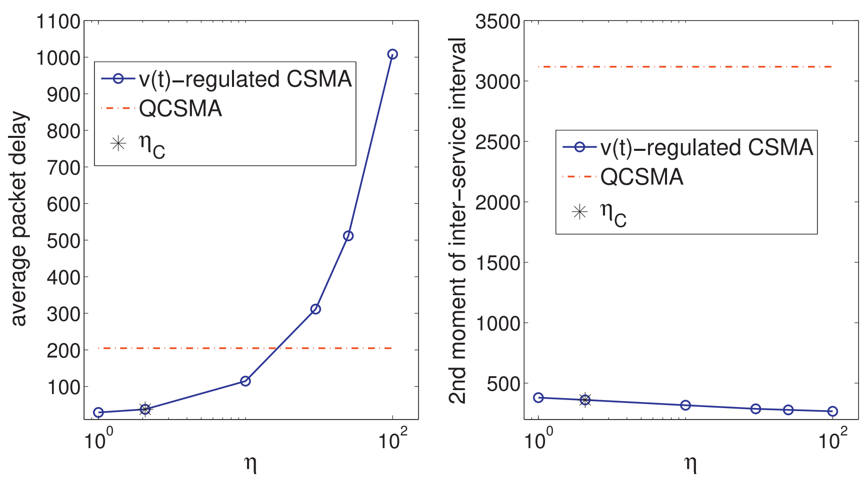

Fig. 8. Numerical performance of delay and 2nd moment of inter-service interval with respect to threshold η under topology in Fig. 5, given arrival rate $\lambda = 0.1$.

note this decrease in J_l is trivial compared to the increase in delay when η becomes large. Therefore, we consider η_C to be a good choice for the thresholds.

5.2. Numerical comparison in a grid topology

It is pointed out in [11] that regular throughput-optimal CSMA algorithms (such as the QCSMA algorithm) suffer from serious temporal starvation in grid/lattice topology. In this subsection, we show that under a grid topology, our proposed $\mathbf{v}(t)$-regulated CSMA algorithm does not suffer as much from the temporal starvation as the QCSMA algorithm. In fact, our proposed algorithm far outperforms the QCSMA algorithm in both delay and second moment of inter-service intervals. Note that different from the algorithm in [11] which is intended for grid/lattice topology only, our proposed algorithm works for arbitrary network topologies under general conflict graph interference model.

Specifically, we employ a two-hop interference model under the grid topology in Fig. 9. The arrival rate vector is set as $\lambda_l = \frac{1}{4}\lambda$ for $l = 1, 4$; $\lambda_l = \frac{1}{6}\lambda$, for $l = 5, 9, 14, 18, 23, 25, 27, 28, 31$; and $\lambda_l = \frac{1}{12}\lambda$, otherwise. The stabilizable range for λ is $0 < \lambda < 1$, since these arrival rates can be represented by a convex combination of 12 maximal matching schedules. The arrival processes $A_l(t)$ follow independent Bernoulli processes with parameter λ_l, $\forall l \in \mathcal{L}$. In addition, we let $\epsilon_2 = \frac{1}{\lambda} - 1$ which denotes the distance between the current arrival rates and the maximum per-link throughput.

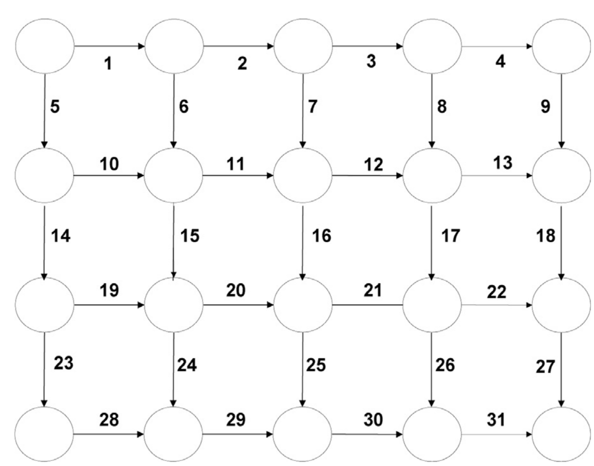

Fig. 9. Grid network topology for simulation comparison.

Similar to the results under the 10-link topology in the previous section, results in Fig. 10 indicate that the proposed algorithm

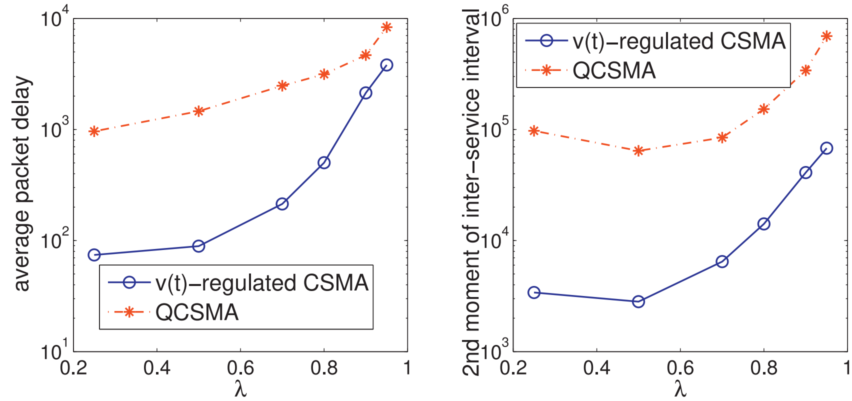

Fig. 10. Numerical performance comparison of delay and 2nd moment of inter-service interval under grid topology in Fig. 9, with $\eta_l = \eta_C$, $\forall l \in \mathcal{L}$.

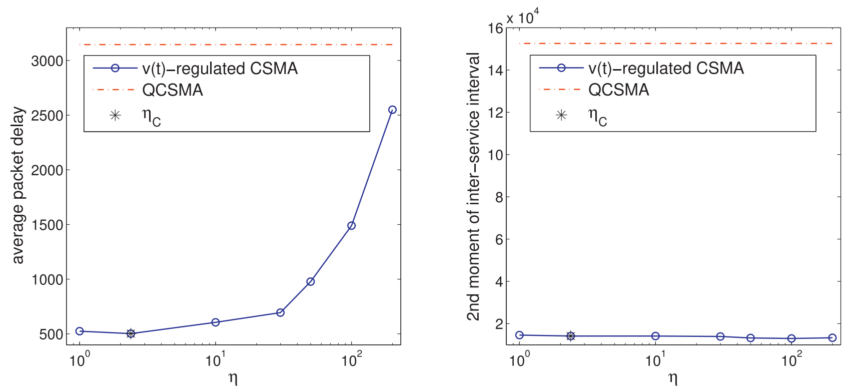

Fig. 11. Numerical performance of delay and 2nd moment of inter-service interval with respect to threshold η under grid topology in Fig. 9, when $\lambda = 0.9$.

significantly outperforms the QCSMA algorithm in terms of delay and second moment of inter-service intervals. In Fig. 11, we show that η_C, determined by (18), is a suitable choice for the threshold, since a larger threshold can lead to a large increase in delay with insignificant improvement on the second moment of inter-service intervals.

6. Conclusions and future works

In this paper, we proposed a $\mathbf{v}(t)$-regulated CSMA algorithm that can be implemented via a distributed method and achieve optimal throughput in wireless networks with single-hop transmissions. In the algorithm, link scheduling is performed favoring links with sufficiently large queue lengths to reduce average delay and ensure a more frequent switch between schedules. Via both hardware implementation and numerical evaluations, we show that compared to the QCSMA algorithm, the proposed algorithm significantly improves the delay performance and mitigates the problem of temporal starvation.

In our future work, we will further study the performance of the proposed CSMA algorithm via a larger scale testbed implementation. Proving the throughput optimality without the time-scale separation assumption will also be one of our future research directions.

Acknowledgment

A preliminary version of this paper has appeared in the Allerton Conference 2012 [17]. This work has been supported in part by NSF under grant number CCF-0914912.

Appendix A. Proof of Theorem 2

We utilize the following two lemmas to prove Theorem 2.

Lemma 4. *Given $kT \le t < (k+1)T$ and any $\epsilon_4 > 0$, the following inequality holds*

$$(1+\epsilon_4) \max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(kT))} \sum_{l\in\mathbf{x}} w_l(t) \ge \max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(t))} \sum_{l\in\mathbf{x}} w_l(t),$$

when

$$||\mathbf{w}(t)|| > \frac{1}{\epsilon_4} \max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} f(f^{-1}(\eta_l) + (T-1)A_M). \quad (19)$$

Proof. Let $t = kT + i$, $0 \le i < T$. We can bound $Q_l(t)$ by the following

$$Q_l(kT) - i \le Q_l(t) \le Q_l(kT) + iA_M, \quad (20)$$

according to the queue dynamics (2). For analytical simplicity, we denote the following independent sets

$$\mathbf{x}_1 \triangleq \arg \max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(kT))} \sum_{l\in\mathbf{x}} w_l(t),$$

$$\mathbf{x}_2 \triangleq \arg \max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(t))} \sum_{l\in\mathbf{x}} w_l(t).$$

Since

$$\mathbf{x}_2 = (\mathbf{x}_2 \cap \mathcal{L}(\mathbf{v}(kT))) \cup (\mathbf{x}_2 \backslash \mathcal{L}(\mathbf{v}(kT)))$$

and

$$\sum_{l\in\mathbf{x}_2\cap\mathcal{L}(\mathbf{v}(kT))} w_l(t) \le \sum_{l\in\mathbf{x}_1} w_l(t),$$

we obtain that

$$\sum_{l\in\mathbf{x}_2} w_l(t) - \sum_{l\in\mathbf{x}_1} w_l(t) \le \sum_{l\in\mathbf{x}_2\setminus\mathcal{L}(\mathbf{v}(kT))} w_l(t) \le \sum_{l\in\mathbf{x}_2\setminus\mathcal{L}(\mathbf{v}(kT))} f(f^{-1}(\eta_l)+(T-1)A_M), \tag{21}$$

where we have employed (20) in the last inequality.

When (19) holds,

$$||\mathbf{w}(t)|| > \max_{l\in\mathcal{L}} f(f^{-1}(\eta_l)+(T-1)A_M),$$

which leads to the following inequality:

$$\max_{l\in\mathcal{L}} Q_l(t) > \max_{l\in\mathcal{L}} f^{-1}(\eta_l)+(T-1)A_M \ge Q_j(kT)+(T-1)A_M \ge Q_j(t),$$

for any $j\in\mathcal{L}\setminus\mathcal{L}(\mathbf{v}(kT))$. Hence, when (19) holds, $\arg\max_{l\in\mathcal{L}} Q_l(t)\in\mathcal{L}(\mathbf{v}(kT))$ and

$$\sum_{l\in\mathbf{x}_1} w_l(t) \ge \max_{l\in\mathcal{L}(\mathbf{v}(kT))} w_l(t) = ||\mathbf{w}(t)||. \tag{22}$$

When (19) is satisfied, we conclude

$$(1+\epsilon_4)\sum_{l\in\mathbf{x}_1} w_l(t) \ge \sum_{l\in\mathbf{x}_1} w_l(t) + \epsilon_4||\mathbf{w}(t)|| > \sum_{l\in\mathbf{x}_1} w_l(t) + \max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} f(f^{-1}(\eta_l)+(T-1)A_M) \ge \sum_{l\in\mathbf{x}_2} w_l(t),$$

where the first and the last inequalities follow (22) and (21), respectively. This completes the proof of Lemma 4. □

Lemma 5. *With the $\mathbf{v}_T(t)$-regulated CSMA algorithm, under the time-scale separation assumption, for any given $0<\epsilon_1', \delta_1<1$, we can find $B'(\epsilon_1',\delta_1)$[2], such that, whenever, $||\mathbf{w}(t)||>B'$,*

$$\sum_{l\in\mathcal{L}} w_l(t)\mu_l(t) \ge \left(1-\frac{\epsilon_1'}{2}\right)\max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(t))}\sum_{l\in\mathbf{x}} w_l(t). \tag{23}$$

Proof. Similar to the derivation of inequality (11) in Lemma 1, under the time-scale separation assumption, we can obtain inequality (24) through the regular throughput-optimal CSMA algorithm on $\mathbf{v}_T(t)$-regulated topology $(\mathcal{N}, L(\mathbf{v}_T(t)))$. Specifically, given $0<\epsilon_1', \delta_1<1$ and $\epsilon_1=\frac{\epsilon_1'^2}{4}$, we can find $B_1(\epsilon_1,\delta_1)>0$ such that w.p. greater than $(1-\delta_1)$, the $\mathbf{v}_T(t)$-regulated CSMA algorithm schedules $(\mu_l(t))_{l\in\mathcal{L}}$ satisfy:

$$\sum_{l\in\mathcal{L}} w_{l,T}(t)\mu_l(t) \ge \left(1-\frac{\epsilon_1}{2}\right)\max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(kT))}\sum_{l\in\mathbf{x}} w_{l,T}(t), \quad \text{whenever } \max_{l\in\mathcal{L}} w_{l,T}(t) > B_1. \tag{24}$$

Since $f'(x)\le 1$, $\forall x\ge 0$ (from Property (iii) of the link weight function f), and

$$f(Q_l(t)) - f(Q_l(kT)) = f'(Q_l')(Q_l(t)-Q_l(kT))$$

where Q_l' lies between $Q_l(t)$ and $Q_l(kT)$, from (20), we have

$$w_l(kT) - i \le w_l(t) \le w_l(kT) + iA_M, \forall l\in\mathcal{L}. \tag{25}$$

From (25) we obtain that

$$\sum_{l\in\mathbf{x}_1} w_l(t) \le \sum_{l\in\mathbf{x}_1}(w_l(kT)+iA_M) \le \max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(kT))}\sum_{l\in\mathbf{x}} w_l(kT) + iA_M\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{\{l\in\mathbf{x}\}}, \tag{26}$$

and

$$\sum_{l\in\mathcal{L}} w_l(t)\mu_l(t) \ge \sum_{l\in\mathcal{L}} w_l(kT)\mu_l(t) - i\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{\{l\in\mathbf{x}\}}. \tag{27}$$

Applying (26) and (27) to (24), we have, w.p. greater than $(1-\delta_1)$, whenever $||\mathbf{w}(kT)||>B_1$,

$$\begin{aligned}\sum_{l\in\mathcal{L}} w_l(t)\mu_l(t) &\ge (1-\frac{\epsilon_1}{2})\max_{\mathbf{x}\in\mathcal{I}(\mathbf{v}(kT))}\sum_{l\in\mathbf{x}} w_{l,T}(t) - i\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{\{l\in\mathbf{x}\}}\\ &\ge (1-\frac{\epsilon_1}{2})\sum_{l\in\mathbf{x}_1} w_l(t) - (T-1)\left(1+A_M(1-\frac{\epsilon_1}{2})\right)\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{\{l\in\mathbf{x}\}}\\ &\ge (1-\epsilon_1)\sum_{l\in\mathbf{x}_1} w_l(t), \text{ when } ||\mathbf{w}(t)||>B_3,\\ &= (1-\sqrt{\epsilon_1})(1+\sqrt{\epsilon_1})\sum_{l\in\mathbf{x}_1} w_l(t)\end{aligned} \tag{28}$$

where

$$B_3 \triangleq \max\left\{f\left(f^{-1}(B_1)+A_M(T-1)\right), \frac{2}{\epsilon_1}(T-1)\left[1+A_M(1-\frac{\epsilon_1}{2})\right]\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{\{l\in\mathbf{x}\}}\right\}.$$

Note that when $||\mathbf{w}(t)||>B_3$,

$$||\mathbf{w}(kT)|| \ge f(||\mathbf{Q}(t)||-iA_M) > f(f^{-1}(B_3)-(T-1)A_M) \ge B_1,$$

where we have utilized inequality (20).

Employing Lemma 4 to (28) with $\epsilon_4=\sqrt{\epsilon_1}$, we conclude, w.p. greater than $(1-\delta_1)$, whenever $||\mathbf{w}(t)||>B'$,

$$\sum_{l\in\mathcal{L}} w_l(t)\mu_l(t) \ge \left(1-\frac{\epsilon_1'}{2}\right)\sum_{l\in\mathbf{x}_2} w_l(t),$$

where $B'=\max\{B_3, \frac{2}{\epsilon_1'}\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} f(f^{-1}(\eta_l)+(T-1)A_M)\}$, which completes the proof. □

Since the $\mathbf{v}_T(t)$-regulated CSMA algorithm has the same property ((23) in Lemma 5) as the $\mathbf{v}(t)$-regulated CSMA algorithm ((6) in Lemma 1), the proof of Theorem 2 directly follows that of Theorem 1 in Section 2.5.

[2] The construction of such a constant B' is provided in the proof.

References

[1] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Trans. Autom. Control 37 (12) (1992) 1936–1948.
[2] L. Georgiadis, M. Neely, L. Tassiulas, Resource allocation and cross-layer control in wireless networks, Found. Trends Netw. (2006) 1–149.
[3] L. Jiang, J. Walrand, A distributed CSMA algorithm for throughput and utility maximization in wireless networks, in: Proceedings 46th Annual Allerton Conference on Communication, Control and Computing, 2008, pp. 1511–1519.
[4] J. Ni, B. Tan, R. Srikant, Q-CSMA: queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks, in: Proc. IEEE INFOCOM 2010 Mini-Conference, 2010, pp. 1–5.
[5] S. Rajagopalan, D. Shah, J. Shin, Network adiabatic theorem: an efficient randomized protocol for contention resolution, in: SIGMETRICS'09: Proceeding of the eleventh international joint conference on measurement and modeling of computer systems, 2009, pp. 133–144.
[6] J. Ghaderi, R. Srikant, On the design of efficient CSMA algorithms for wireless networks, in: Proc. IEEE conference on Decision and Control (CDC 2010), 2010, pp. 954–959.
[7] A. Proutiere, Y. Yi, T. Lan, M. Chiang, Resource allocation over network dynamics without timescale separation, in: Proc. IEEE INFOCOM 2010 Mini-Conference, 2010, pp. 1–5.
[8] L. Jiang, J. Walrand, Convergence and stability of a distributed CSMA algorithm for maximal network throughput, in: IEEE CDC'09, 2009, pp. 4840–4845.
[9] L. Jiang, M. Leconte, J. Ni, R. Srikant, J. Walrand, Fast mixing of parallel glauber dynamics and low-delay CSMA scheduling, in: IEEE INFOCOM 2011 Mini-Conference, 2011, pp. 371–375.

[10] K. Lam, C. Chau, M. Chen, S. Liew, Mixing time and temporal starvation of general CSMA networks with multiple frequency agility? in: Proc. of 2012 IEEE International Symposium on Information Theory (ISIT 2012), 2012, pp. 2676–2680.
[11] M. Lotfinezhad, P. Marbach, Throughput-optimal random access with order-optimal delay, in: Proc. IEEE INFOCOM 2011, 2011, pp. 2867–2875.
[12] D. Xue, R. Murawski, E. Ekici, Distributed utility-optimal scheduling with finite buffers, in: Proc. WiOpt 2012, 2012, pp. 278–285.
[13] M. Neely, Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varing Channels, Mass. Inst. Technol. (MIT), Cambridge, MA, 2003 Ph.d. Dissertation.
[14] S. Meyn, R. Tweedie, Criteria for stability of Markovian process I: discrete time chains, in: Advances in Applied Probability, vol. 24, 1992, pp. 542–574.
[15] S. Kim, H. Kim, J. Song, M. Yu, P. Mah, Nanoqplus : a multi-threaded operating system with memory protection mechanism for WSNs, in: Proc. CKWSN 2008, 2008, pp. 1–8.
[16] Texas Instruments, MSP430 Microcontroller: http://focus.ti.com/docs/prod/folders/print/msp430f1611.html.
[17] D. Xue, E. Ekici, On reducing delay and temporal starvation of queue-length-based CSMA algorithms, in: Proc. 50th Annual Allerton Conference on Communication, Control, and Computing, 2012, pp. 754–761.

Dongyue Xue received the B.S. degree in information engineering from Shanghai Jiaotong University, Shanghai, China, in 2009, and the M.S. and Ph.D. degrees in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2013. His research interests include cross-layer scheduling in wireless networks and dynamic resource allocation in cognitive radio networks.

Eylem Ekici received the B.S. and M.S. degrees in computer engineering from Bogaziçi University, Istanbul, Turkey, in 1997 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology in 2002. He is currently a professor with the Department of Electrical and Computer Engineering, The Ohio State University. His current research interests include cognitive radio networks, vehicular communication systems, and nextgeneration wireless systems, with a focus on algorithm design, medium access control protocols, resource man agement, and analysis of network architectures and protocols. He is the TPC Co-Chair of the IEEE INFOCOM 2017. He is an IEEE Fellow, an associate editor of the IEEE Transactions on Mobile Computing and Computer Networks. He is a former associate editor of the IEEE/ACM Transactions on Networking.

Rania Ibrahim has received her B.Sc. and M.Sc. from Computer and Systems Engineering Department, Alexandria University, Egypt in 2012 and 2015 respectively. She has also received MASc. from Electrical and Computer Engineering Department, University of Waterloo, Canada in 2016. Currently, she is a Ph.D. student at Computer Science Department, Purdue University, USA in the machine learning group. Her research interests include Machine Learning, Data Mining, Bioinformatics and Networks.

Moustafa Youssef is the founder & director of the Wireless Research Center at Egypt-Japan University of Science and Technology (E-JUST). His research interests include mobile computing, location determination technologies, pervasive computing, mobile wireless networks, and network security. He is an associate editor for the ACM TSAS, a previous area editor of the ACM MC2R and served on the organizing and technical committees of numerous prestigious conferences. Prof. Youssef is the recipient of the 2003 University of Maryland Invention of the Year award, the 2010 TWAS-AAS-Microsoft Award for Young Scientists, the 2012 Egyptian State Award, the 2013 and 2014 COMESA Innovation Awards, the 2013 ACM SIGSpatial GIS Conference Best Paper Award, among many others. He is also an ACM Distinguished Speaker and an ACM Distinguished Scientist.