# On Reducing Delay and Temporal Starvation of Queue-Length-Based CSMA Algorithms

Dongyue Xue, Eylem Ekici

Department of Electrical and Computer Engineering
Ohio State University, USA
Email: {xued, ekici}@ece.osu.edu

***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. We prove that the proposed algorithm is throughput optimal, and show through numerical evaluations that the algorithm indeed mitigates the temporal starvation problem and achieves far better delay performance than other throughput-optimal CSMA algorithms.**

## I. 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 queue-length-based CSMA algorithms have been proposed in the literature [4]-[7] that achieve throughput optimality, which we refer to as *regular throughput-optimal CSMA algorithms* in the following discussion. Although these CSMA algorithms have been proved to be throughput-optimal [5][6], they suffer from the following problems: (1) *Temporal starvation*, defined in [11] 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 [5] [13]. 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 [10] 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 [11]. Congestion control using virtual queues has been proposed in [13] in an attempt 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. 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 [4][5] and verified in [8][9]. Furthermore, our proposed CSMA algorithm is shown via simulations to have a much more favorable delay performance than a regular throughput-optimal CSMA algorithm [5] 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 II. Further discussions are provided in Section III. Specifically, a method to approach time-scale separation is presented in Section III-A, and we provide a guideline on choosing thresholds for the proposed algorithm in Section III-B. We present the numerical results in Section IV and conclude our work in Section V.

## II. $\mathbf{v}(t)$-Regulated CSMA Algorithm

We introduce the network model in Section II-A, with the proposal and theoretical performance analysis of the $\mathbf{v}(t)$-regulated CSMA algorithm presented in Sections II-B and II-C, respectively.

### A. Network Model

Consider a time-slotted 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 [4]-[7]. 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 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 $\lambda_l$.[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:

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

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

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:

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

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 $E$.

### B. $\mathbf{v}(t)$-Regulated CSMA Algorithm and Its Distributed Implementation

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 $\eta_l$, 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.

In the following, we introduce definitions necessary for our CSMA algorithm. We first define the indicator variable

$$v_l(t) = \mathbf{1}_{\{w_l(t) > \eta_l\}}, \ \forall l \in \mathcal{L},$$

where $w_l(t)$ is the link weight of $l \in \mathcal{L}$ and $\eta_l$ 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 $\eta_l$ (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:

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

i.e., $\mathcal{L}(\mathbf{v}(t))$ is the set of links whose link weights $w_l(t)$ are greater than the corresponding thresholds $\eta_l$. 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:

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

**$\mathbf{v}(t)$-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:

$$\sum_{\mathbf{x}(t)\in\mathcal{I}(\mathbf{v}(t))} p_{\mathbf{x}(t)} = 1 \text{ and } \cup_{p_{\mathbf{x}(t)}>0}\, \mathbf{x}(t) = \mathcal{L}(\mathbf{v}(t)). \quad (2)$$

2. Scheduling link rates:
   - 2.1 $\forall l \in \mathbf{x}(t)$,
     - 2.1.1 If $\sum_{j\in\mathcal{N}_l\setminus\{l\}} \mu_j(t-1)v_j(t) = 0$: $\mu_l(t) = 1$, 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 $\forall l \in \mathcal{L}\setminus\mathbf{x}(t)$, $\mu_l(t) = \mu_l(t-1)v_l(t)$.

Fig. 1. $\mathbf{v}(t)$-regulated CSMA algorithm

In Figure 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:

**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., $\sum_{j\in\mathcal{N}_l\setminus\{l\}} \mu_j(t-1)v_j(t) = 0$, then link $l$ is scheduled service ($\mu_l(t) = 1$) with link activation probability

$$p_l(t) \triangleq \frac{e^{w_l(t)}}{1+e^{w_l(t)}}.$$

Otherwise, $\mu_l(t) = 0$.

**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 [4]; $f(x) = \log\log(x+e)$ in [6]; $f(x) = \log(x+1)$ in [5][7]. 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):* $f'(x) \le 1$, $\forall x \ge 0$.

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 $\eta_l$) 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 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 suffer 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.

In [16], 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 II-C), we prove the throughput-optimality of the proposed algorithm. Numerical analysis on delay performance and the temporal starvation issue is provided in Section IV.

### C. 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. 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:

**Case i:** $l \in \mathbf{y}(t) \cap \mathbf{x}(t)$, where $\mathbf{x}(t)$ is the independent set chosen according to Step 1 in Figure 1. In this case, $\mu_j(t-1)v_j(t) = 0$, $\forall j \in \mathcal{N}_l\setminus\{l\}$, according to Step 2.1. Given any $j \in \mathcal{N}_l\setminus\{l\}$, we know that $j \notin \mathbf{x}(t)$, since $\mathbf{x}(t)$ is an independent set and $l \in \mathbf{x}(t)$. If $v_j(t) = 0$, then $\mu_j(t) = 0$ according to Step 2.2. Otherwise (i.e., when $v_j(t) = 1$), $\mu_j(t-1) = 0$, and hence $\mu_j(t) = \mu_j(t-1) = 0$ according to Step 2.2. Therefore, $j \notin \mathbf{y}(t)$.

**Case ii:** $l \in \mathbf{y}(t)\setminus\mathbf{x}(t)$. In this case, $\mu_l(t-1) = 1$ and $v_l(t) = 1$ according to Step 2.2. For any given $j \in \mathcal{N}_l\setminus\{l\}$, we have

$$\sum_{k\in\mathcal{N}_j\setminus\{j\}} \mu_k(t-1)v_k(t) \ge \mu_l(t-1)v_l(t) = 1. \quad (3)$$

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 (3) and Step 2.1.2. Otherwise (i.e., when $j \notin \mathbf{x}(t)$), from Step 2.2, $\mu_j(t) \le \mu_j(t-1) = 0$. 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 $j \notin \mathbf{y}(t)$, $\forall j \in \mathcal{N}_l\setminus\{l\}$. ■

To support our analysis of the throughput performance, we introduce two related lemmas, Lemma 1 and Lemma 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 $\epsilon_1$ and $\delta_1$ 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, whenever $||\mathbf{w}(t)||_\infty > B$,

$$\sum_{l\in\mathcal{L}} w_l(t)\mu_l(t) \geq (1-\epsilon_1)\max_{\mathbf{x}\in\mathcal{I}} \sum_{l\in\mathbf{x}} w_l(t), \tag{4}$$

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 (4), $\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t)$ can be considered as the maximal weight matching over all feasible schedulers.

*Proof:* The proof is provided in Appendix A. ∎

We define the capacity region $\Lambda$ 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 $\Lambda$, 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 $\Lambda$, 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][14], so we omit the proof of Lemma 2 for brevity.

The throughput optimality is concluded in Theorem 1.

*Theorem 1:* The $\mathbf{v}(t)$-regulated CSMA algorithm is *throughput-optimal*, i.e., the packet queues are stable in the mean [3][7], for any arrival rate vector strictly within the capacity region $\Lambda$:

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\{\left[\sum_{l\in\mathcal{L}} f^2(Q_l(t))\right]^{\frac{1}{2}}\right\} \leq \frac{B_2}{\epsilon}, \tag{5}$$

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

$$B_2 \triangleq Lf(A_M)+Lf(Q_M)+Lf(Q_M+A_M)A_M+\gamma BL, \tag{6}$$

$$\epsilon \triangleq \min_{l\in\mathcal{L}}[(\gamma(1+\epsilon_2)-1)\lambda_l - \epsilon_3 A_M] > 0,$$

with $\gamma \triangleq (1-\epsilon_1)(1-\delta_1)$. Note that $\epsilon_1$, $\delta_1$, $B$ are defined in Lemma 1; $\epsilon_2$ defined in Lemma 2; and $\epsilon_3$, $Q_M$ defined in Property (ii) of the link weight function $f$.

*Proof:* The proof of Theorem 1 is given in Appendix B. ∎

Since $\epsilon > 0$ is required to ensure a positive upper-bound in (5), we must have from definition of $\epsilon$ that

$$\gamma > \frac{1}{1+\epsilon_2}, \tag{7}$$

and

$$\epsilon_3 < \min_{l\in\mathcal{L}}\frac{[\gamma(1+\epsilon_2)-1]\lambda_l}{A_M}. \tag{8}$$

Since $\gamma < 1$ can be chosen arbitrarily close to 1 (due to the fact that $\epsilon_1$ and $\delta_1$ 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 $\gamma$ and $\epsilon_3$ such that the inequalities (7) and (8) hold.

Note that the underlying Markov chain is positive recurrent due to the queue stability in the mean (5) according to [15], which implies the stability of the network [10].

## III. 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 III-A and a feasible choice of thresholds is introduced in III-B.

### A. Approaching Time-Scale Separation

We recall that Lemma 1 (and hence Theorem 1) in Section II-C 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 Figure 2. Specifically, we make the Markov chain of the schedules converge to the steady state distributions by updating the link weights $w_{l,T}(t)$ in the $\mathbf{v}_T(t)$-regulated CSMA algorithm periodically every $T$ time slots, $l \in \mathcal{L}$,

$$w_{l,T}(t) = f(Q_l(kT)),\ kT \leq t < (k+1)T, \tag{9}$$

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

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

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.

*Proof:* The proof is provided in [16]. ∎

### B. 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.,

**$\mathbf{v}_T(t)$-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 $\forall l \in \mathcal{L}\setminus\mathbf{x}(t)$, $\mu_l(t) = \mu_l(t-1)v_{l,T}(t)$.

Fig. 2. $\mathbf{v}_T(t)$-regulated CSMA algorithm

[10]), 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 (5) can be considered as an upper-bound for packet queue lengths, which is an indicator on the delay performance. Thus, we choose the thresholds $\eta_l$, $l \in \mathcal{L}$, 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 $\epsilon_1 = \delta_1 \leq \frac{\epsilon_2}{2(1+\epsilon_2)}$ [2],
$$\eta_l = \eta_C \triangleq \frac{(L+1)\log 2 + \log\frac{1+\epsilon_2}{\epsilon_2}}{2\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{l\in\mathbf{x}}}, l \in \mathcal{L}, \qquad (10)$$
guarantees that $\frac{B_2}{\epsilon}$ is minimized.

*Proof:* According to the definition of $B_2$ in (6) and the definition of $B$ in Lemma 1, it is sufficient to prove that the choice (10) ensures that
$$\frac{2}{\epsilon_1}\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}\eta_l \leq B_1 = \frac{1}{\epsilon_1}\left(L\log 2 + \log\frac{1}{\delta_1}\right).$$
For any $l \in \mathcal{L}$, we obtain from (10) that
$$\frac{2}{\epsilon_1}\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}}\eta_l = \frac{1}{\epsilon_1}\left((L+1)\log 2 + \log\frac{1+\epsilon_2}{\epsilon_2}\right) \leq B_1,$$
completing the proof. ■

If local links do not have the knowledge of $\epsilon_2$, 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 $\eta_l$:
$$\eta_l = \frac{(L+1)\log 2}{2\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathcal{L}}\mathbf{1}_{l\in\mathbf{x}}}, l \in \mathcal{L}.$$

We note again that the value $\eta_C$ 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 IV, we show that choosing the threshold as $\eta_C$ indeed leads to favorable delay performance and significantly mitigates the temporal starvation compared to a regular throughput-optimal CSMA algorithm.

[2]Since $\epsilon_1$ and $\delta_1$ can be chosen arbitrarily small, the given choice of $\epsilon_1$ and $\delta_1$ is feasible and ensures that the constraint (7) holds, i.e, $\gamma = (1-\delta_1)^2 > 1 - 2\delta_1 \geq \frac{1}{1+\epsilon_2}$.

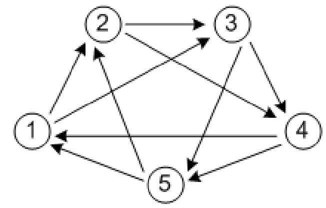


Fig. 3. 10-link network topology for numerical evaluations

## IV. Numerical Results

In this section, we present a numerical comparative study of $\mathbf{v}(t)$-regulated CSMA algorithm, vis-a-vis the QCSMA algorithm [5], 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 [5][7] that the $\log(\cdot)$ form yields better delay performance than $f(x) = x$ [4] and $f(x) = \log\log(x+e)$ [6]. In the following two subsections, we illustrate the performance using two different interference models under two different network topologies.

### A. Numerical Evaluation in a 10-Link Topology

In this subsection, we employ a node-exclusive model (or one-hop interference model) under the 10-link topology in Figure 3. 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 under the node-exclusive model, the stabilizable range of $\lambda$ is $0 < \lambda < 0.2$, with $\epsilon_2 = \frac{0.2}{\lambda} - 1$ characterizing the distance between $\lambda$ and the maximum per-link throughput 0.2. The arrival processes $A_l(t)$ follow independent Bernoulli processes with parameter $\lambda$.

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 4 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 Figure 4. 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.

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 $i$-th packet of link $l \in \mathcal{L}$, we define the second moment of

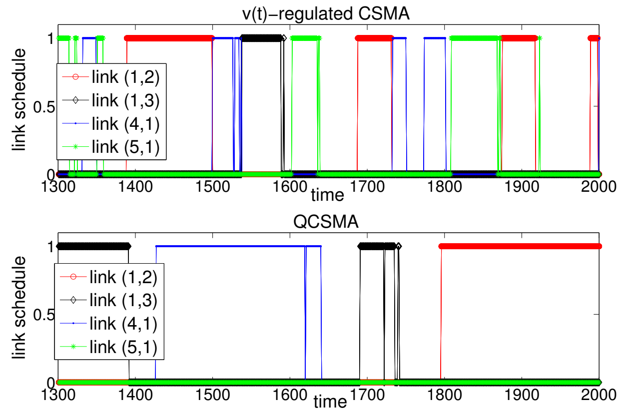


Fig. 4. Snapshot of Link schedules when arrival rate $\lambda = 0.19$ under topology in Figure 3

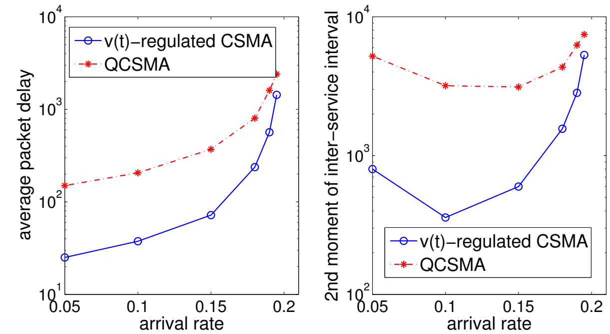


Fig. 5. Performance of delay and 2nd moment of inter-service intervals with respect to arrival rate under topology in Figure 3, given $\eta_l = \eta_C$, $\forall l \in \mathcal{L}$

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. Due to the symmetric simulation topology, in the following we refer to the second moment of inter-service intervals as the average of $J_l$ over all links $l \in \mathcal{L}$.

Figure 5 illustrates the delay and the second moment of inter-service intervals under the two algorithms, with $\eta_l = \eta_C$, $\forall l \in \mathcal{L}$. 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 almost one order of magnitude 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 $\eta_C$ ($\eta_C = 2.7$ in this simulation 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 $\lambda$ is close to is $0.2$. 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 $65\%$ smaller under the $\mathbf{v}(t)$-regulated CSMA algorithm than the QCSMA algorithm; and the second moment of inter-service intervals is $55\%$ 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 Figure 5. 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.

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 Figure 6, we show the performance of delay and second moment of inter-service intervals by varying the value of $\eta$, where we let $\eta_l = \eta$, $l \in \mathcal{L}$. The starred point in Figure 6 denotes the case with $\eta_l = \eta_C$, $l \in \mathcal{L}$. We observe that $\eta_C$ 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 $\eta$ grows larger than $\eta_C$. However, we note that the decrease in $J_l$ is trivial compared to the increase in delay as $\eta$ becomes larger.

From Figure 6, 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 $\eta$ 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 $\eta$ 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 $\eta$ 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-$\eta$-regime.

We also observe that, when $\eta$ 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 Figure 6. However, we note this decrease in $J_l$ is trivial compared to the increase in delay when $\eta$ becomes large. Therefore, we consider $\eta_C$ to be a good choice for the thresholds.

### *B. Simulation Comparison in a Grid Topology*

It is pointed out in [12] that regular throughput-optimal CSMA algorithms (such as the QCSMA algorithm) suffer from serious temporal starvation in grid/lattice topology.

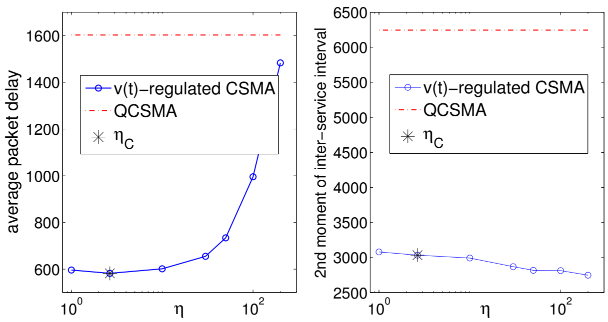


Fig. 6. Performance of delay and 2nd moment of inter-service interval with respect to threshold $\eta$ under topology in Figure 3, given arrival rate $\lambda = 0.19$

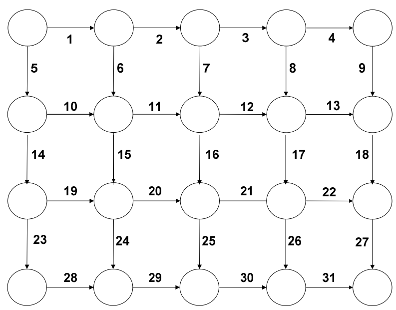

Fig. 7. Grid network topology for simulation comparison

In [16], we show that under a grid topology in Figure 7, 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 [12] which is intended for grid/lattice topology only, our proposed algorithm works for arbitrary network topologies under general conflict graph interference model.

## V. 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 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 testbed implementations. Proving the throughput optimality without time-scale separation assumption will also be one of our future research directions.

## References


[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", in *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[2] L. Georgiadis, M. Neely and L.Tassiulas, "Resource allocation and cross-Layer control in wireless networks", in *Foundations and Trends in Networking*, pp. 1-149, 2006.

[3] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels", in *IEEE/ACM Transactions on Networking*, pp. 411-424, Vol. 13, No.2, April 2005.

[4] L. Jiang and 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*, September 2008.

[5] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks", in *Proc. IEEE INFOCOM 2010 Mini-Conference*, April 2010.

[6] S. Rajagopalan, D. Shah, and 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*, pp. 133-144, 2009.

[7] J. Ghaderi and R. Srikant, "On the design of efficient CSMA algorithms for wireless networks", in *Proc. IEEE conference on Decision and Control (CDC 2010)*, December, 2010.

[8] A. Proutiere, Y. Yi, T. Lan, and M. Chiang, "Resource allocation over network dynamics without timescale separation", in *Proc. IEEE INFOCOM 2010 Mini-Conference*, March 2010.

[9] L. Jiang and J. Walrand, "Convergence and stability of a distributed CSMA algorithm for maximal network throughput", in *IEEE CDC'09*, 2009.

[10] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling", in *IEEE INFOCOM 2011 Mini-Conference*, April 2011.

[11] K. Lam, C. Chau, M. Chen, and 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)*, July, 2012.

[12] M. Lotfinezhad and P. Marbach, "Throughput-optimal random access with order-optimal delay", in *Proc. IEEE INFOCOM 2011*, April 2011.

[13] D. Xue, R. Murawski, and E. Ekici, "Distributed utility-optimal scheduling with finite buffers", in *Proc. WiOpt 2012*, May 2012.

[14] M. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varing channels", Ph.D. dissertation, Mass. Inst. Technol. (MIT), Cambridge, MA, 2003.

[15] S. Meyn and R. Tweedie, "Criteria for stability of Markovian process I: discrete time chains", in *Advances in Applied Probability*, vol. 24, pp. 542-574, 1992.

[16] D. Xue and E. Ekici, "On reducing delay and temporal starvation of queue-length-based CSMA algorithms", Technical Report, Ohio State University, available: http://www.ece.osu.edu/∼xued/regulated.pdf


## Appendix A
## Proof of Lemma 1

We have shown in [16] that the $\mathbf{v}(t)$-regulated CSMA algorithm is equivalent to the scheduler in Figure 8, where Step 1 and Step 2 of the equivalent scheduler (in Figure 8) form the regular throughput-optimal CSMA algorithm [10] with respect to the $\mathbf{v}(t)$-regulated topology $(\mathcal{N}, \mathcal{L}(\mathbf{v}(t)))$. According to Proposition 2 in [5] under the time-scale separation assumption, 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) schedules $(\mu_l(t))_{l \in \mathcal{L}}$ satisfying, whenever $||\mathbf{w}(t)|| > B_1$,

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

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

**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. 8. An equivalent scheduler for the proof in Lemma 1

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

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

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

$$(1-\frac{\epsilon_1}{2})\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t)$$
$$<\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t) - \max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t)(1-v_l(t))$$
$$\le\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t)v_l(t).$$

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

$$\sum_{l\in\mathcal{L}} \mu_l(t)w_l(t)$$
$$\ge(1-\frac{\epsilon_1}{2})\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t)v_l(t)$$
$$>(1-\frac{\epsilon_1}{2})^2\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t) \ge (1-\epsilon_1)\max_{\mathbf{x}\in\mathcal{I}}\sum_{l\in\mathbf{x}} w_l(t),$$

which concludes the proof of Lemma 1.

## APPENDIX B
## PROOF OF THEOREM 1

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

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

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

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

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, following the analysis in [16] (where we have employed Lemma 1 and substituted the stationary randomized algorithm STAT in Lemma 2), we obtain an upper-bound for the expectation of the Lyapunov drift $\mathbb{E}\{\Delta(t)\}$ in (12):

$$\mathbb{E}\{\Delta(t)\}$$
$$\le B_2 + \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(Q_l(t))\left[\epsilon_3 A_M + A_l(t) - \gamma\mu_l^{STAT}(t)\right]\right\} \quad (13)$$
$$=B_2 + \sum_{l\in\mathcal{L}} \mathbb{E}\left\{f(Q_l(t))\left[\epsilon_3 A_M + \lambda_l - \gamma\lambda_l(1+\epsilon_2)\right]\right\}$$
$$\le B_2 - \epsilon\sum_{l\in\mathcal{L}} \mathbb{E}\{f(Q_l(t))\},$$

where we have employed Lemma 2 to (13).

Taking the time-average over $t = 0, 1, ..., T-1$ of both sides of (13) and taking the limsup with respect to $T$, we conclude

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

which proves (5).