

# Optimal CSMA-based Wireless Communication with Worst-case Delay and Non-uniform Sizes

Hongxing Li and Nitin Vaidya

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA

Email: {hxli, nhv}@illinois.edu

**Abstract**—Carrier Sense Multiple Access (CSMA) protocols have been shown to reach the full capacity region for data communication in wireless networks, with polynomial complexity. However, current literature achieves the throughput optimality with an exponential delay scaling with the network size, even in a simplified scenario for transmission jobs with uniform sizes. Although CSMA protocols with order-optimal average delay have been proposed for specific topologies, no existing work can provide *worst-case delay guarantee* for each job in *general network settings*, not to mention the case when the jobs have *non-uniform lengths* while the throughput optimality is still targeted. In this paper, we tackle on this issue by proposing a two-timescale CSMA-based data communication protocol with dynamic decisions on rate control, link scheduling, job transmission and dropping in polynomial complexity. Through rigorous analysis, we demonstrate that the proposed protocol can achieve a throughput utility arbitrarily close to its offline optima for jobs with non-uniform sizes and worst-case delay guarantees, with a tradeoff of longer maximum allowable delay.

## I. INTRODUCTION

The efficacy of a wireless communication algorithm can be examined with three criteria: high throughput, low response delay and low computation/communication complexity. However, it is commonly accepted that there is a tradeoff among the three dimensions of algorithm performances [17]. Maximum-weight scheduling (MWS) [19] algorithms are proven to be throughput-optimal, however incurring exponential computation complexity as the network size grows up. Low-complexity algorithms ([10] and references therein) are proposed to approximate the MWS, while achieving only a fraction of the optimal throughput.

CSMA-style random access control protocols have been studied intensely in recent years for its low complexity and provable optimality in throughput maximization [5], [15]. Nevertheless, it comes with an exponentially long delay scaling with the network size [12]. Although some recent efforts [1]–[3], [6], [12], [16], [18] try to improve the delay performance and have even achieved asymptotic bounds on the average delay [12], [16], [18] in specific topologies, the worst-case delay guarantee, which is a more practical concern in real-world implementations ensuring that each transmission job is either served or dropped before its maximum allowable delay, is yet to be studied. The difficulty further escalates

This research is supported in part by National Science Foundation awards CNS-11-17539. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

if we still aim to obtain (close-to-)optimal throughput utility at the same time, with general network topologies and low computation/communication complexities.

Apart from above, a common assumption is shared by current literature such that each transmission job has the same size and is packed in a single data unit, *e.g.*, one data packet, which can be completely delivered within one time slot. This idealized model fails to capture the diverse job sizes of some mainstream applications. For example, one *Twitter* update may need just tens of bytes while a video clip on *Youtube* may be in the size of several mega-bytes. A more practical model should allow the existence of transmission jobs consisted of one/multiple consecutive data packets, which should either be fully delivered to the destination or completely dropped. Partial reception of the transmission job brings no utility to the network, *e.g.*, a video clip with missing information may not be decodable. When coupled with the worst-case delay guarantee, *i.e.*, each transmission job instead of one packet is either delivered or dropped before its service deadline, we should explore novel designs for the low-complexity throughput-optimal CSMA protocol.

In this paper, we investigate the throughput-utility optimal CSMA protocol in general network topologies with low computation/communication complexities and worst-case delay guarantees for transmission jobs with diverse sizes. A two-timescale algorithm is proposed to dynamically make decisions in each time slot on: 1) *rate control*: how many jobs should be admitted into the network such that congestion could be avoided while the throughput utility is maximized? 2) *link scheduling*: which subset of the links should be simultaneously scheduled for transmission such that no collision will occur while the network capacity can be fully exploited? 3) *job transmission*: how many jobs, from each category of job sizes and worst-case delay requirements, should be transmitted over the scheduled links? 4) *job dropping*: how many jobs of each category should be dropped so as to meet the worst-case delay bounds? A CSMA-style random access control mechanism is integrated with the Lyapunov optimization framework [13] for the algorithm design. To be specific, the link scheduling is carried out with the CSMA protocol and *randomly* generates collision-free transmissions, while the rate control, and job transmission and dropping decisions are *deterministically* made based on the network status in each time slot. Rigorous analysis demonstrates that our protocol can achieve a throughput utility, which can be made arbitrarily close to its optima, with polynomial computation/communication complexity at

each link and guaranteed worst-case delay for jobs with non-uniform sizes, at a tradeoff of longer maximum allowable delay.

The contribution of this paper is summarized as follows,

▷ To our best knowledge, we are the first to investigate the existence of worst-case delay guarantees and non-uniform job sizes for CSMA protocols in general network topologies.

▷ A CSMA-based two-timescale wireless communication algorithm is proposed to dynamically decide the rate control, link scheduling, and job transmission and dropping in each time slot, with an objective to maximize the time-averaged throughput utility.

▷ Theoretical analysis demonstrates that our proposed algorithm can guarantee the worst-case delay for all job sizes, and achieve a throughput-utility that can be arbitrarily close to its optimality, with a polynomial computation/communication complexity at each link and the tradeoff of a longer maximum allowable delay.

The rest of the paper is organized as follows. We discuss the related works in Sec. II and present the problem model in Sec. III. The two-timescale dynamic algorithm is introduced with details in Sec. IV. In Sec. V, we rigorously analyze the efficiency of our algorithm. Finally, we conclude this paper in Sec. VI.

## II. RELATED WORK

CSMA protocols have attracted tremendous attention in recent years [5], [15] mainly due to its potential to simultaneously achieve high throughput and low complexity, and its implementation in a distributed fashion.

With perfect and instantaneous carrier sensing assumption (no collision will happen), Jiang *et al.* [5] introduce a continuous-time CSMA protocol, that can achieve the optimal throughput. A discrete-time queue-length based CSMA protocol is next proposed by Ni *et al.* [15] to reach the full capacity region, explicitly considering the avoidance of collisions without the perfect carrier sensing assumption.

However, it has been shown that it is hard to achieve throughput optimality and low delay simultaneously with CSMA protocols [17]. Hence, a rich body of research efforts have been devoted to decreasing the delay for CSMA protocols. Shah *et al.* [16] present a CSMA protocol with order-optimal delay for networks with geometry. Jiang *et al.* [3] demonstrate the relation between the small mixing time and a low delay, and investigate on how to tighten the generic bound of mixing time for specific topologies. For networks with bounded interference degree and an arrival rate within only a fraction of the capacity region, Jiang *et al.* [2] show that the average delay grows polynomially with the network size under parallel Glauber dynamics, and a constantly bounded mean delay independent of the network size is proved by Subramanian *et al.* [18]. In contrast, Lotfinezhad *et al.* [12] achieve not only the throughput optimality but also an order-optimal delay, in the torus topology. However, it is not clear

whether the above improvements can be extended to *general network topologies*.

Lee *et al.* [7] examine the delay performance of a class of CSMA protocol by tuning the control parameters. Nevertheless, there is no evidence on whether tuning parameters could fundamentally improve the exponential order of delay performance. Lam *et al.* [6] try to improve the average delay with multiple physical channels, however, each link can be scheduled on at most one channel at a time, which cannot fully exploit the capacity region. Huang *et al.* [1] explore the power of multiple virtual channels to reduce the head-of-line delay, with a definition different from the average delay. Throughput utility, instead of queue lengths, is used as the scheduling weight. Order-optimal head-of-line delay can be obtained in a close-loop setting with rate control.

The only work that considers the deadline for data transmission by Li *et al.* in [8]. However, the solution proposed by [8] only applies to a complete graph (each link collides with each other), but cannot be adapted to general topologies.

Different from the papers discussed above, this paper practically considers the guarantee of *worst-case delay* bounds and the communication service for jobs with *non-uniform sizes*. Meanwhile, the throughput-utility optimality can still be achieved with a *low complexity*, for *general topologies*.

## III. PROBLEM MODEL

We have a wireless network composed of a node set  $\mathcal{N}$  and link set  $\mathcal{E}$ . Each source-destination pair is within one hop distance, which means each source just needs exactly one transmission to reach its destination without relaying. Each link has unit-capacity, *i.e.*, transmitting at most one packet in one time slot.

We consider a general interference model by defining an interference-relation set  $\mathcal{C}_i$  for each link  $i \in \mathcal{E}$ . Each link  $j \in \mathcal{C}_i$  will cause collision to link  $i$  scheduled concurrently.

Different from existing efforts on throughput-optimal CSMA protocols assuming identical sizes of transmission jobs, we model the diverse job sizes of various network applications by differentiating types of transmission jobs. Let  $\mathcal{M}$  denote the set of job types. For each job type  $m \in \mathcal{M}$ , it is composed of  $s_m$  consecutive data packets, which should be either fully delivered to its destination or entirely dropped.

The network runs in a time-slotted fashion. In each time slot  $t \geq 0$ , a random number of  $A_{mi}(t)$  ( $\forall m \in \mathcal{M}, i \in \mathcal{E}$ ) jobs arrive at the transmitter of link  $i$ . Here,  $A_{mi}(t)$  is i.i.d. in  $[0, A_{mi}^{max}]$  with  $A_{mi}^{max}$  as the maximum job arrival rate for type  $m$  job at link  $i$ . Uncontrolled admission of job arrivals may cause congestion in the network. Thus, a rate control decision  $r_{mi}(t)$  should be made such that jobs of type- $m$  are admitted into the job queue on link  $i$  with

$$r_{mi}(t) \in [0, A_{mi}(t)], \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (1)$$

### A. Job queues

After jobs of type  $m \in \mathcal{M}$  are admitted to the source of link  $i \in \mathcal{E}$ , they are injected into a queue  $Q_{mi}(t)$  of unsent jobs with queueing law as follows,

$$Q_{mi}(t+1) = \max\{Q_{mi}(t) - \mu_{mi}(t) - d_{mi}(t) \times s_m, 0\} + r_{mi}(t) \times s_m, \quad \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (2)$$

Here, the length of  $Q_{mi}(t)$  is the total number of packets waiting to be delivered at time slot  $t$ .  $d_{mi}(t)$  is the number of type  $m$  jobs that are dropped by link  $i$  at time slot  $t$ , as a result of meeting its delay deadline (to be introduced shortly), with

$$d_{mi}(t) \in [0, d_{mi}^{max}], \quad \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (3)$$

where,  $d_{mi}^{max}$  is the maximum dropping rate.  $\mu_{mi}(t)$  is the number of type  $m$  packets delivered over link  $i$  at time slot  $t$ . Since unit-capacity is assumed for each link, we have that

### B. Link scheduling and job transmission

Each link  $i \in \mathcal{E}$  is indicated to be either active (transmitting) or idle in each time slot with binary variable  $x_i$  as follows,

$$x_i(t) = \begin{cases} 1 & \text{if link } i \text{ is scheduled in slot } t, \forall i \in \mathcal{E}. \\ 0 & \text{Otherwise.} \end{cases} \quad (4)$$

A feasible link schedule should ensure that no pair of mutually interfering links can be active concurrently, *i.e.*,

$$x_i(t) + x_j(t) \leq 1, \quad \forall j \in \mathcal{C}_i, i \in \mathcal{E}. \quad (5)$$

If link  $i$  is active in slot  $t$ , it needs to decide which type of jobs should be served with the available capacity. Recall that each link has unit capacity, at most one type of jobs can be served in current slot with the following capacity constraint,

$$\sum_{m \in \mathcal{M}} \mu_{mi}(t) = x_i(t), \quad \forall i \in \mathcal{E}. \quad (6)$$

$$\mu_{mi}(t) \in \{0, 1\}, \quad \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (7)$$

### C. Worst case delay guarantee

As stated previously, we novelly address the worst-case delay bound for each admitted job in the network as follows,

Each type- $m$  job for link  $i$  is either scheduled for transmission or dropped (subject to a penalty) before its maximum delay  $D_m$ ,

$$\forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (8)$$

It is natural that a penalty,  $\beta > 0$ , for each dropped packet should be charged, such that it is not rational for each link  $i$  to greedily admit jobs for now while to drop them later.

### D. Useful definitions

We present some important definitions that will be used in the rest of the paper.

*Definition 1 (Queue and Network Stability [13]):* A queue  $Q$  is *strongly stable* (or *stable* for short) if and only if

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(Q(\tau)) < \infty,$$

where  $Q(\tau)$  is the queue size at time slot  $\tau$  and  $\mathbb{E}(\cdot)$  is the expectation. A network is *strongly stable* (or *stable* for short) if and only if all queues in the network are strongly stable.

*Theorem 1 (Necessity & Sufficiency for Queue Stability [13]):* For any queue  $Q$  with the following queuing law,

$$Q(t+1) = Q(t) - \gamma(t) + \alpha(t),$$

where  $\alpha(t)$  and  $\gamma(t)$  are the arrival and departure rates in time slot  $t$ , respectively, the following results hold:

$\mathcal{N}$	Set of nodes	$\mathcal{E}$	Set of links
$\mathcal{M}$	Set of job types	$\mathcal{C}_i$	Collision set of link $i$
$\mathbb{E}(\cdot)$	The expectation	$U(\cdot)$	Utility function
$s_m$	Size of type- $m$ jobs		
$s^{max}$	Maximum job size of all types		
$D_m$	Worst-case delay of type- $m$ jobs		
$A_{mi}(t)$	Arrival rate of type- $m$ jobs on link $i$ in time slot $t$		
$A_{mi}^{max}$	Maximum arrival rate of type- $m$ jobs on link $i$		
$r_{mi}(t)$	Admitted type- $m$ jobs on link $i$ in time slot $t$		
$\eta_{mi}(t)$	Auxiliary variable for $r_{mi}(t)$ in time slot $t$		
$x_i(t)$	Binary var: link $i$ is scheduled in time slot $t$ ?		
$\mu_{mi}(t)$	Binary var: type- $m$ job is transmitted over link $i$ in time slot $t$ ?		
$\mu_{mi}(t^-)$	Binary var: type- $m$ job has not finished transmission over link $i$ in time slot $t$ ?		
$d_{mi}(t)$	# of dropped type- $m$ jobs on link $i$ in time slot $t$		
$d_{mi}^{max}$	Maximum drop rate of type- $m$ jobs on link $i$		
$Q_{mi}(t)$	Packet queue of type- $m$ jobs on link $i$ in time slot $t$		
$Y_{mi}(t)$	Rate control virtual queue for type- $m$ jobs on link $i$ at time $t$		
$Z_{mi}(t)$	Delay virtual queue for type- $m$ jobs on link $i$ at time $t$		
$\epsilon_{mi}$	Constant for delay virtual queue $Z_{mi}(t)$		
$V$	User-defined positive constant in dynamic algorithm		
$B$	Quantity defined in Sec. IV		

TABLE I  
LIST OF NOTATIONS.

*Necessity:* If queue  $Q$  is strongly stable, then its average incoming rate  $\bar{\alpha} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(\alpha(\tau))$  is no larger than the average outgoing rate  $\bar{\gamma} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(\gamma(\tau))$ .

*Sufficiency:* If the average incoming rate  $\bar{\alpha}$  is strictly smaller than the average outgoing rate  $\bar{\gamma}$ , *i.e.*,  $\bar{\alpha} + \epsilon \leq \bar{\gamma}$  with  $\epsilon > 0$ , then queue  $Q$  is strongly stable.

Hereinafter, for any variable  $\alpha(t)$ , we denote its time-averaged value as  $\bar{\alpha}$ , *i.e.*,  $\bar{\alpha} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(\alpha(\tau))$ .

### E. Throughput utility maximization problem

Our objective is to dynamically decide the rate control, link scheduling, and job transmission and dropping, such that the time-averaged net utility (throughput utility minus the job dropping penalty) can be maximized while the worst-case delay is guaranteed for jobs with non-uniform sizes.

$$\max \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} [U(\bar{r}_{mi} \times s_m) - \beta \bar{d}_{mi} \times s_m] \quad (9)$$

*s.t.* Network stability, and Constraint (1),(3),(7),(4),(5),(6),(8).

Here,  $U(\cdot)$  is the throughput utility function, which is non-negative, non-decreasing, concave and differentiable. It is reasonable to have  $\beta > U'(0)$  such that admitting one job into the queue for now while dropping it later brings no positive utility gain.

Important notations are summarized in Table I.

## IV. CSMA-BASED DYNAMIC WIRELESS COMMUNICATION ALGORITHM

In this section, we introduce our CSMA-based wireless communication algorithm, which dynamically decides the rate control, link scheduling, and job transmission and dropping, so as to maximize the time-averaged throughput utility as defined in (9). We first define two important virtual queues (to deal with rate control and delay bounds, respectively), and then present the algorithm design in details by solving four one-slot optimization problems in each time slot in order to approximate (9).

### A. Virtual queues

We have two types of virtual queues to assist the algorithm design.

**Virtual queue for rate control:** To deal with the case when the utility function  $U(\cdot)$  is non-linear [13], each link  $i \in \mathcal{E}$  has the following virtual queue for its rate control on each job type  $m \in \mathcal{M}$ ,

$$Y_{mi}(t+1) = \max\{Y_{mi}(t) - r_{mi}(t) \cdot s_m, 0\} + \eta_{mi}(t) \cdot s_m \quad \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (10)$$

Here,  $\eta_{mi}(t)$  is an auxiliary variable with

$$\eta_{mi}(t) \in [0, A_{mi}^{max}], \quad \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (11)$$

The rationale is that, if virtual queue  $Y_{mi}(t)$  is kept stable, we have  $\bar{\eta}_{mi} \leq \bar{r}_{mi}$  with Theorem 1, *i.e.*, the time-averaged value of  $\eta_{mi}(t) \cdot s_m$  constitutes a lower bound for the average throughput. Later on, we will show that maximizing the utility of  $\bar{\eta}_{mi} \cdot s_m$  can approximately maximize the utility of average throughput  $\bar{r}_{mi} \cdot s_m$ .

**Virtual queue for delay bound:** The  $\epsilon$ -persistence queue [14]<sup>1</sup> is applied in order to meet the QoS constraint. For each job type  $m \in \mathcal{M}$ , each link  $i \in \mathcal{E}$  maintains the following virtual queue,

$$Z_{mi}(t+1) = \max\{Z_{mi}(t) + \mathbf{1}_{\{Q_{mi}(t) > 0\}}(\epsilon_{mi} - \mu_{mi}(t)) - d_{mi}(t) \times s_m - \mathbf{1}_{\{Q_{mi}(t) = 0\}}, 0\}, \quad \forall m \in \mathcal{M}, i \in \mathcal{E}. \quad (12)$$

Here,  $\mathbf{1}_{\{\cdot\}}$  is a binary indicator function.  $\epsilon_{mi}$  is a positive constant. The virtual queue  $Z_{mi}(t)$  approximately keeps track of the delay information for data packet queue  $Q_{mi}(t)$  and assists our algorithm design (to be introduced shortly).

### B. Distributed dynamic algorithm

We derive the dynamic algorithm by decoupling the time-averaged utility maximization problem (9) into four one-slot optimization problems to be solved in each time slot.

Each link  $i \in \mathcal{E}$  maintains a set of queues  $\Theta(t) = \{Y_{mi}(t), Q_{mi}(t), Z_{mi}(t) | \forall m \in \mathcal{M}, i \in \mathcal{E}\}$ . We define the Lyapunov function as follows,

$$L(\Theta(t)) = \frac{1}{2} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} [(Y_{mi}(t))^2 + (Q_{mi}(t))^2 + (Z_{mi}(t))^2]. \quad (13)$$

The one-slot conditional Lyapunov drift is

$$\Delta(\Theta(t)) = L(\Theta(t+1)) - L(\Theta(t)). \quad (14)$$

By squaring the queueing laws in Eqn. (2), (10) and (12), We can have the *drift-plus-penalty* inequality as follows (derivation details are included in technical report [9]),

$$\begin{aligned} & \Delta(\Theta(t)) - V \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\eta_{mi}(t) \cdot s_m) - \beta d_{mi}(t) \times s_m) \\ & \leq B + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} Z_{mi}(t) \cdot \epsilon_{mi} - \Phi_1(t) - \Phi_2(t) - \Phi_3(t) - \Phi_4(t). \end{aligned} \quad (15)$$

Here,  $B = \frac{1}{2} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} [3(A_{mi}^{max} \cdot s_m)^2 + 2(1 + d_{mi}^{max} \cdot s_m)^2 + (\epsilon_{mi})^2]$  is a constant value, and  $V > 0$  is a user-defined parameter to adjust the weight of net utility in the expression.  $\Phi_1(t)$ ,  $\Phi_2(t)$ ,  $\Phi_3(t)$  and  $\Phi_4(t)$  are as follows,

- Terms related to auxiliary variables  $\eta_{mi}(t)$ :

$$\Phi_1(t) = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} [V \cdot U(\eta_{mi}(t) \cdot s_m) - Y_{mi}(t) \cdot \eta_{mi}(t) \cdot s_m].$$

- Terms related to rate control variables  $r_{mi}(t)$ :

$$\Phi_2(t) = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} r_{mi}(t) \cdot s_m \cdot [Y_{mi}(t) - Q_{mi}(t)].$$

- Terms related to link scheduling and job transmission variables  $\mu_{mi}(t)$ :

$$\Phi_3(t) = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mu_{mi}(t) \cdot [Q_{mi}(t) + Z_{mi}(t)].$$

- Terms related to packet drop variables  $d_{mi}(t)$ :

$$\Phi_4(t) = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} d_{mi}(t) \cdot s_m \cdot [Q_{mi}(t) + Z_{mi}(t) - V \cdot \beta].$$

According to *Lyapunov optimization theory* [13], we can maximize a lower bound of the time-averaged throughput utility and find optimal solutions to the rate control, link scheduling, job transmission and dropping variables by minimizing the RHS of the *drift-plus-penalty* equality (15), observing the queue lengths  $\Theta(t)$  and the packet arrival  $A_{mi}(t)$  in each time slot  $t$ . Hence, we propose a dynamic algorithm to solve the one-slot optimization problem in each time slot  $t$  as follows,

$$\begin{aligned} & \max \quad \Phi_1(t) + \Phi_2(t) + \Phi_3(t) + \Phi_4(t) \\ & \text{s.t.} \quad \text{Constraints (1),(3),(7),(4),(5),(6),(11)}. \end{aligned} \quad (16)$$

Note that, the delay constraint (8) is not included in the one-slot optimization, since it could be satisfied by the stability of virtual queue  $Z_{mi}(t)$  to be shown in Sec. V.

The maximization problem in (16) can be decoupled into four independent optimization problems:

$$\begin{aligned} & \max \quad \Phi_1(t) \\ & \text{s.t.} \quad \text{Constraint (11)}, \end{aligned} \quad (17)$$

which is related to the optimal decision on the auxiliary variable  $\eta_{mi}(t)$ ; and

$$\begin{aligned} & \max \quad \Phi_2(t) \\ & \text{s.t.} \quad \text{Constraint (1)}, \end{aligned} \quad (18)$$

which is related to the optimal decision on the rate control variable  $r_{mi}(t)$ ; and

$$\begin{aligned} & \max \quad \Phi_3(t) \\ & \text{s.t.} \quad \text{Constraint (7),(4),(5),(6)}, \end{aligned} \quad (19)$$

which is related to the optimal decision on the link scheduling variable  $x_i(t)$  and job transmission variable  $\mu_{mi}(t)$ ; and

$$\begin{aligned} & \max \quad \Phi_4(t) \\ & \text{s.t.} \quad \text{Constraint (3)}, \end{aligned} \quad (20)$$

which is related to the optimal decision on the job dropping variable  $d_{mi}(t)$ . Hence, we have the following dynamic algorithm with optimal solutions to each variable.

<sup>1</sup>Note that, in [14], the  $\epsilon$ -persistence queue can only handle the case when transmission jobs have the uniform size. In this paper, we adapt this technique to the jobs with non-uniform sizes.

1) *Rate control*: We solve (17) and (18) to decide the auxiliary variables and rate control variables ( $\forall m \in \mathcal{M}, i \in \mathcal{E}$ ) as follows,

$$\eta_{mi}(t) = \max\{\min\{U'^{-1}(\frac{Y_{mi}(t)}{V})/s_m, A_{mi}^{max}\}, 0\}, \quad (21)$$

where,  $U'^{-1}(\cdot)$  is the reverse function of the first-order derivative of the utility function; and

$$r_{mi} = \begin{cases} A_{mi}(t) & \text{if } Y_{mi}(t) - Q_{mi}(t) > 0 \\ 0 & \text{Otherwise.} \end{cases} \quad (22)$$

**Remark**: Virtual queue  $Y_{mi}(t)$  can be regarded as the unused tokens for data admission. A large value for  $Y_{mi}(t)$  indicates adequate available tokens, which results in fewer new tokens, i.e.,  $\eta_{mi}(t)$ , to be added in this slot. Meanwhile,  $Q_{mi}(t)$  reflects the congestion level on the link.  $Y_{mi}(t) - Q_{mi}(t) > 0$  means we have enough tokens while relatively low congestion. Thus, we admit all the arrived jobs. Otherwise, no job is admitted into the network.

2) *Link scheduling and job transmission with CSMA*: We design the following mechanism to approximate the optimal solution to (19). Our CSMA-based scheduling mechanism runs in a two-timescale fashion: super slot and regular slot. Each super slot is composed of  $T \geq s^{max}$  ( $s^{max} = \max_{m \in \mathcal{M}}\{s_m\}$ ) regular slots, while each regular slot has the same definition as in [15] and our problem model. The link scheduling decisions are made upon the beginning of each super slot and remain fixed throughout each regular slot in that super slot. However, the served job-types are decided in every regular slot dynamically. To be specific, we have that

- If  $t = nT$  with  $n \geq 0$ : this is the beginning of the  $n^{th}$  super slot. The regular slot of this type is composed of two consecutive phases: control phase and scheduling phase.
  - *Control phase*: In this phase, all the links distributively randomly generate a collision-free *control schedule*  $\mathbf{z}(t) = [z_1, \dots, z_i(t), \dots, z_E(t)]$  with  $z_i(t) \in \{0, 1\}$ ,  $\forall i \in \mathcal{E}$ . This *control schedule* is not the final decision on link scheduling, but indicates the links which may make changes to its scheduling decision in the *scheduling phase*. The *control phase* has  $W$  mini-slots<sup>2</sup>. At the start of this phase, each link  $i$  uniformly randomly select an integer  $T_i$  in  $[0, W - 1]$  and backoff for  $T_i$  mini-slots. Link  $i$  has the following possible actions:
    - ▷ If link  $i$  hears no ‘INTENT’ message before the  $(T_i + 1)$ th mini-slot, it broadcasts an ‘INTENT’ message at mini-slot  $T_i + 1$ .
      - If there is no collision, link  $i$  is included in the *control schedule* and we have  $z_i(t) = 1$ .
      - Otherwise, link  $i$  is not selected into the *control schedule* and we have  $z_i(t) = 0$ .
    - ▷ If link  $i$  hears any ‘INTENT’ message before the  $(T_i + 1)$ th mini-slot, it is not included in the *control schedule* and we have  $z_i(t) = 0$ .

<sup>2</sup>Compared with the regular slot, the length of mini-slots is negligible.

– *Scheduling phase*: We make the link scheduling decisions based on the *control scheduling* and the link scheduling decisions in previous regular slot as follows:

▷ If  $z_i(t) = 0$ ,  $x_i(t) = x_i(t - 1)$ .

▷ If  $z_i(t) = 1$ , we further have that

– If there is any active link in link  $i$ 's collision set, i.e.,  $\exists j \in \mathcal{C}_i, x_j(t - 1) = 1$ , link  $i$  is not scheduled in this regular slot and  $x_i(t) = 0$ .

– Otherwise, link  $i$  randomly becomes active in this regular slot with probability  $p_i = \frac{e^{w_i(t)}}{1 + e^{w_i(t)}}$ , i.e.,

$$\begin{cases} x_i(t) = 1 & \text{with probability } p_i \\ x_i(t) = 0 & \text{with probability } \bar{p}_i = 1 - p_i. \end{cases} \quad (23)$$

Here, weight  $w_i(t) = \max_{m \in \mathcal{M}}\{Q_{mi}(t) + Z_{mi}(t)\}$ .

- If  $t = nT + \tau$  with  $n \geq 0$  and  $\tau \in (0, T - 1]$ : this regular slot is within the  $n^{th}$  super slot. We keep the link scheduling decision made in slot  $nT$ . However, each link can decide which job type is served in this slot:
  - If link  $i$  is not scheduled in slot  $nT$ , i.e.,  $x_i(nT) = 0$ , it keeps inactive in slot  $t$  with  $x_i(t) = 0$  and  $\mu_{mi}(t) = 0, \forall m \in \mathcal{M}$ .
  - If the transmission job  $m^* \in \mathcal{M}$  scheduled in previous slot is not finished, link  $i$  goes on with transmitting job type  $m^*$  with  $x_i(t) = 1, \mu_{m^*i}(t) = 1$  and  $\mu_{mi}(t) = 0, \forall m \in \mathcal{M}, m \neq m^*$ . We use  $\mu_{mi}(t^-) \in \{0, 1\}$  to indicate whether the previously scheduled job is finished or not, with  $\mu_{mi}(t^-) = 1$  for unfinished case while  $\mu_{mi}(t^-) = 0$  for cases where either jobs are completed delivered or no job is scheduled in previous slot.
  - Otherwise, link  $i$  is still active in slot  $t$ . However, it will select the job type, with maximum weight, to be served in this slot:

$$m^* = \arg \max_{m \in \mathcal{M}, (n+1)T - t \geq s_m} \{w_{mi}(t)\}. \quad (24)$$

Here,  $(n + 1)T - t \geq s_m$  ensures that the selected job can be served before the end of this super slot. Thus, at the beginning of the next super slot, there is no on-going unfinished transmission jobs.

3) *Job drop*: In each time slot  $t$ , we deterministically decide the number of time-out jobs to be dropped, by solving (20), as follows,

$$d_{mi}(t) = \begin{cases} d_{mi}^{max} & \text{if } Q_{mi}(t) + Z_{mi}(t) > V\beta \\ 0 & \text{Otherwise} \end{cases}. \quad (25)$$

**Remark**: The rationale is that, each link is reluctant to drop packets until the queue lengths exceed certain threshold, above which we may indicate that packets are suffering a long delay.

The dynamic algorithm is summarized in Alg. 1.

### C. Computation/communication complexities

In each time slot, Algorithm 1 incurs polynomial computation and communication complexities at each link  $i \in \mathcal{E}$ .

---

**Algorithm 1** Dynamic Net Utility Maximization Algorithm in Time Slot  $t$ 


---

**Input:**  $Q_{mi}(t)$ ,  $Y_{mi}(t)$ ,  $Z_{mi}(t)$ ,  $A_{mi}(t)$ ,  $A_{mi}^{max}$ ,  $d_{mi}^{max}$ , ( $\forall m \in \mathcal{M}$ ,  $i \in \mathcal{E}$ ),  $V$ ,  $U(\cdot)$  and  $\beta$ .

**Output:**  $\eta_{mi}(t)$ ,  $r_{mi}(t)$ ,  $x_i(t)$ ,  $\mu_{mi}(t)$ ,  $d_{mi}(t)$ , ( $\forall m \in [1, M]$ ).

- 1: **Rate Control:** For each job-type  $m$ , link  $i$  decides the data admission rate  $r_{mi}(t)$  and auxiliary variable  $\eta_{mi}(t)$  by Eqn. (22) and (21), respectively.
  - 2: **Link Scheduling:** Each link  $i$  distributively execute the CSMA algorithm as in Alg. 2 and find solutions to  $x_i(t)$  and  $\mu_{mi}(t)$ .
  - 3: **Job Dropping:** For each job-type  $m$ , link  $i$  decides the job dropping rate  $d_{mi}(t)$  Eqn. (25).
  - 4: Update queues  $Q_{mi}(t+1)$ ,  $Y_{mi}(t+1)$  and  $Z_{mi}(t+1)$  based on queuing law (2), (10) and (12), respectively.
- 

---

**Algorithm 2** CSMA Scheduling Algorithm at Link  $i$  in Time Slot  $t$ 


---

**Input:**  $Q_{mi}(t)$ ,  $Z_{mi}(t)$  and  $\mu_{mi}(t^-)$ , ( $\forall m \in \mathcal{M}$ ).

**Output:**  $x_i(t)$  and  $\mu_{mi}(t)$ , ( $\forall m \in \mathcal{M}$ ).

**If**  $(t \bmod T) = 0$ :

- 1: Uniformly randomly choose an integer  $T_i$  from  $[1, W]$ , and wait for  $T_i$  mini-slots;
  - If** link  $i$  hears an ‘INTENT’ message from any link in  $\mathcal{C}_i$  before the  $(T_i + 1)$ th mini-slot, link  $i$  is not included in  $\mathbf{z}(t)$  and  $z_i(t) := 0$ . No ‘INTENT’ message will be sent by  $i$ ;
  - Else** Link  $i$  broadcasts an ‘INTENT’ message to all links in  $\mathcal{C}_i$  at the beginning of the  $(T_i + 1)$ th mini-slot;
  - If** there is a collision, link  $i$  is not included in  $\mathbf{z}(t)$ . Set  $z_i(t) := 0$ .
  - Else**, link  $i$  is included in  $\mathbf{z}(t)$  by setting  $z_i(t) := 1$ .
- 2: **If**  $z_i(t) = 0$ , set  $x_i(t) := x_i(t-1)$ ;
- 3: **Else**,
  - If** no link in  $\mathcal{C}_i$  was active in slot  $t-1$
  - Set  $x_i(t) := 1$  with probability  $p_i = \frac{e^{w_i(t)}}{1+e^{w_i(t)}}$ ;
  - Or, set  $x_i(t) := 0$  with probability  $\bar{p}_i = 1 - p_i$ .
  - Else**, set  $x_i(t) := 0$ .
- 4: **If**  $x_i(t) = 1$ , set  $\mu_{m^*i}(t) := 1$  with  $m^* = \max_{m \in \mathcal{M}} \{Q_{mi}(t) + Z_{mi}(t)\}$  and  $\mu_{m' i}(t) := 0$  with  $m' \neq m^*$ .
- 5: **Else** set  $\mu_{m' i}(t) := 0$ ,  $\forall m \in \mathcal{M}$ .

**If**  $(t \bmod T) \neq 0$ :

- 1:  $x_i(t) := x_i(t-1)$ .
  - 2: **If**  $x_i(t) = 1$ 
    - If**  $\exists m' \in \mathcal{M}$  with  $\mu_{m' i}(t^-) = 1$ , set  $\mu_{m' i}(t) := 1$  and  $\mu_{m'' i}(t) := 0$  with  $m'' \neq m'$ .
    - Else** set  $\mu_{m^* i}(t) := 1$  with  $m^* = \max_{m \in \mathcal{M}} \{Q_{mi}(t) + Z_{mi}(t)\}$  and  $\mu_{m' i}(t) := 0$  with  $m' \neq m^*$ .
  - 3: **Else** set  $\mu_{m' i}(t) := 0$ ,  $\forall m \in \mathcal{M}$ .
- 

**Computation complexity:** For each job type  $m \in \mathcal{M}$ , link  $i \in \mathcal{E}$  finds the optimal solutions to its auxiliary variable, rate control and job dropping variables in constant time with Algorithm 1. Thus, the overall computation complexity for these variables is in  $O(|\mathcal{M}|)$  for each link in each time slot.

For link scheduling and job transmission, each link consumes constant time on the control schedule, at most  $O(|\mathcal{E}|)$  complexity to check out the scheduling status of mutual interfering links in previous slot, constant time to compute the link scheduling decision, and  $O(|\mathcal{M}|)$  complexity to find the job-type with maximum weight. Hence, the overall complexity

for this part is in  $O(|\mathcal{E}| + |\mathcal{M}|)$ .

To sum up, the computation complexity at each link is in  $O(|\mathcal{E}| + |\mathcal{M}|)$ .

**Communication complexity:** The only communication overhead occurs at the first step of link scheduling at the beginning of each super slot with Algorithm 2. If link  $i$  timeouts before any of its mutual-interfering links in the control phase, it will just broadcasts one ‘INTENT’ message to its neighborhood; otherwise, no message will be sent by link  $i$ . If link  $i$  is included in the control schedule, it takes at most  $O(|\mathcal{E}|)$  communication overhead to find the link scheduling status of its interfering links in previous slot. Therefore, the overall communication complexity for each link is  $O(|\mathcal{E}|)$  in each slot.

## V. PERFORMANCE ANALYSIS

In this section, we present the analytical results of Algorithm 1.

*Lemma 1 (Bounded queue lengths):* Let  $Y_{mi}^{max} = V \cdot U'(0) + A_{mi}^{max} \cdot s_m$ ,  $Q_m^{max} = V \cdot U'(0) + 2A_{mi}^{max} \cdot s_m$  and  $Z_m^{max} = V \cdot \beta / s_m + \epsilon_{mi}$ . If  $d_{mi}^{max} \geq \max\{A_{mi}^{max}, \epsilon_{mi}/s_m\}$ , in each time slot  $t \geq 0$ , the lengths of packet queues and virtual queues are bounded as follows,

$$Q_{mi}(t) \leq Q_{mi}^{max}, Y_{mi}(t) \leq Y_{mi}^{max}, \text{ and } Z_{mi}(t) \leq Z_m^{max}. \quad (26)$$

This lemma is the basis to prove the worst-case delay guarantee in Theorem 2, and can be proved by induction. Details can be found in technical report [9].

*Theorem 2 (Worst-case delay guarantee):* Each job of type  $m \in \mathcal{M}$  on link  $i \in \mathcal{E}$  is either schedule for transmission or dropped before a preset deadline  $D_{mi}$  if we set  $\epsilon_{mi} = \frac{Q_{mi}^{max} + Z_{mi}^{max}}{D_{mi}}$ .

We prove this theorem by contradiction with details in technical report [9].

*Theorem 3 ( $1 - \delta$  weight):* Given any  $\theta$  and  $\delta$  with  $0 < \theta, \delta < 1$ , if  $\max\{\Phi_3(t)\} \geq \frac{1}{\theta}(|\mathcal{E}| \log 2 + \log \frac{1}{\delta})$ , we have that in any beginning timeslot  $nT$  of super frame  $n \geq 0$ , with probability greater than  $1 - \delta$ , Algorithm 2 finds a schedule  $\mu_{mi}(t)$  such that

$$\Phi_3(t) \geq (1 - \theta) \max\{\Phi_3(t)\}. \quad (27)$$

This theorem is proved by glauber dynamics and time separation assumption (commonly assumed in [15] and references therein, and justified by [4], [11]). Details are included in technical report [9]. This theorem will be utilized for the proof to Theorem 4.

*Theorem 4 (Utility-optimality):* The average throughput utility achieved with our proposed Algorithm 1,  $\Psi$ , is within a constant gap  $\frac{(T-1)}{2V} B'' + B' \frac{(T-s^{max}+1)(T-s^{max})}{2VT} + B/V$  from a  $(1-\nu)$  offline optimum,  $\Psi^{*\nu}$ , which uses  $(1-\nu)$  fraction of the full capacity region and has perfect information into the future, as follows,

$$\Psi \geq \Psi^{*\nu} - \frac{(T-1)}{2V} B'' - B' \frac{(T-s^{max}+1)(T-s^{max})}{2VT} - B/V. \quad (28)$$

Here,  $B = \frac{1}{2} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} [3(A_{mi}^{max} s_m)^2 + 2(1 + d_{mi}^{max})^2 + (\epsilon_{mi})^2]$ ,  $B' = \sum_{i \in \mathcal{E}} \sum_{m \in \mathcal{M}} ((A_{mi}^{max} + 2d_{mi}^{max}) \times s_m + 2)$ ,  $B'' = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} ((A_{mi}^{max} s_m)^2 + \epsilon_{mi}^2 + 2d_{mi}^{max} s_m (1 + d_{mi}^{max} s_m))$ , and  $\nu = 1 - \frac{(1-\delta)(1-\theta)(T-s^{max}+1)}{T}$ .

*Proof:*

For each time slot  $t = nT + \tau$  with  $n \geq 0$  and  $\tau \in [0, T - s^{max}]$ , we have that

$$\begin{aligned} & \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) \mu_{mi}(t-1) \\ &= \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) \mu_{mi}(t^-) \\ & \quad + \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) [\mu_{mi}(t-1) - \mu_{mi}(t^-)] \\ &\leq \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) \mu_{mi}(t^-) \\ & \quad + \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) (\mu_{mi}(t) - \mu_{mi}(t^-)) \\ &= \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) \mu_{mi}(t). \end{aligned}$$

The inequality is based on the fact that, in Algorithm 2,  $(\mu_{mi}(t) - \mu_{mi}(t^-))$  is determined by serving the job-types with maximum queue lengths on each link.

Based on the queueing law Eqn. (2) and (12), we have that

$$\begin{aligned} & |Q_{mi}(t) + Z_{mi}(t) - Q_{mi}(t-1) - Z_{mi}(t-1)| \\ &= |r_{mi}(t) \cdot s_m - \mu_{mi}(t) - d_{mi}(t) \cdot s_m \\ & \quad + \mathbf{1}_{\{Q_{mi}(t) > 0\}} (\epsilon_{mi} - \mu_{mi}(t)) - d_{mi}(t) \times s_m - \mathbf{1}_{\{Q_{mi}(t)=0\}}| \\ &\leq (A_{mi}^{max} + 2d_{mi}^{max}) \times s_m + \epsilon_{mi} + 2. \end{aligned}$$

Thus, we further have that,  $\forall i \in \mathcal{E}$ ,

$$\begin{aligned} & \sum_{m \in \mathcal{M}} (Q_{mi}(t-1) + Z_{mi}(t-1)) \mu_{mi}(t-1) \\ &\leq B'_i + \sum_{m \in \mathcal{M}} (Q_{mi}(t) + Z_{mi}(t)) \mu_{mi}(t), \end{aligned} \quad (29)$$

where,  $B'_i = \sum_{m \in \mathcal{M}} ((A_{mi}^{max} + 2d_{mi}^{max}) \times s_m + \epsilon_{mi} + 2)$ .

For time slot  $t = nT + \tau$  with  $\tau \in [0, T - s^{max}]$ , we have the following based on Eqn. (29),

$$\begin{aligned} & \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (Q_{mi}(nT) + Z_{mi}(nT)) \mu_{mi}(nT) \\ &\leq \tau B' + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (Q_{mi}(t) + Z_{mi}(t)) \mu_{mi}(t). \end{aligned} \quad (30)$$

Here,  $B' = \sum_{i \in \mathcal{E}} B'_i$ . The inequality is equivalent to the following

$$\Phi_3(nT) \leq \tau B' + \Phi_3(nT + \tau). \quad (31)$$

Summing up this inequality over time slots with  $\tau \in [0, T - s^{max}]$ , we have that

$$\begin{aligned} & (T - s^{max} + 1) \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (Q_{mi}(nT) + Z_{mi}(nT)) \mu_{mi}(nT) \\ &\leq \frac{(T - s^{max} + 1)(T - s^{max})}{2} B' \\ & \quad + \sum_{\tau \in [0, T - s^{max}]} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (Q_{mi}(nT + \tau) \\ & \quad + Z_{mi}(nT + \tau)) \mu_{mi}(nT + \tau), \end{aligned} \quad (32)$$

which is equivalent to

$$\begin{aligned} & (T - s^{max} + 1) \Phi_3(nT) \\ &\leq \frac{(T - s^{max} + 1)(T - s^{max})}{2} B' + \sum_{\tau \in [0, T - s^{max}]} \Phi_3(nT + \tau), \end{aligned} \quad (33)$$

Based on Theorem 3, we have that, with probability no larger than  $\delta$ , our link scheduling decisions will result in

$$0 \leq \Phi_3(t) < (1 - \theta) \max\{\Phi_3(t)\}, \quad \forall t = nT, n \geq 0. \quad (34)$$

Thus, taking expectations on  $\Phi_3(t)$ , we have that

$$\begin{aligned} \mathbb{E}(\Phi_3(t)) &\geq (1 - \delta)(1 - \theta) \mathbb{E}(\max\{\Phi_3(t)\}) + \delta \cdot 0 \\ &= (1 - \delta)(1 - \theta) \mathbb{E}(\max\{\Phi_3(t)\}), \quad \forall t = nT, n \geq 0. \end{aligned} \quad (35)$$

The above inequality is under the condition that  $t = nT$ . Next, we study the case when  $t = nT + \tau$  with  $\tau \in [0, T - s^{max}]$ .

When  $\tau \in [0, T - s^{max}]$ , we can have the following based on Eqn. (31)

$$\begin{aligned} \mathbb{E}(\Phi_3(nT + \tau)) &\geq -\tau B' + (1 - \delta)(1 - \theta) \mathbb{E}(\max\{\Phi_3(t)\}), \\ &\quad \forall t = nT, n \geq 0. \end{aligned} \quad (36)$$

Since the job arrival is i.i.d., we know that, for any fraction  $1 - \nu$  of the full capacity region, there exists a stationary randomized algorithm solving the rate control, link scheduling and job dropping decisions with offline optimal throughput utility [13]. We denote the optimal solutions, with this stationary randomized algorithm in  $1 - \nu$  capacity region, as  $\eta_{mi}^{*\nu}(t)$ ,  $r_{mi}^{*\nu}(t)$ ,  $x_i^{*\nu}(t)$ ,  $\mu_{mi}^{*\nu}(t)$  and  $d_{mi}^{*\nu}(t)$ , respectively. Let  $\Phi_1^{*\nu}(t)$ ,  $\Phi_2^{*\nu}(t)$ ,  $\Phi_3^{*\nu}(t)$  and  $\Phi_4^{*\nu}(t)$  denote the value of these four expressions under the stationary randomized algorithm. Denote  $\eta_{mi}^{*\nu} = \mathbb{E}\{\eta_{mi}^{*\nu}(t)\}$ ,  $r_{mi}^{*\nu} = \mathbb{E}\{r_{mi}^{*\nu}(t)\}$ ,  $x_i^{*\nu} = \mathbb{E}\{x_i^{*\nu}(t)\}$ ,  $\mu_{mi}^{*\nu} = \mathbb{E}\{\mu_{mi}^{*\nu}(t)\}$  and  $d_{mi}^{*\nu} = \mathbb{E}\{d_{mi}^{*\nu}(t)\}$ .

In Sec. IV, we have seen that our solutions to auxiliary variables, rate control and job-drop decisions maximize the value of  $\Phi_1(t)$ ,  $\Phi_2(t)$  and  $\Phi_4$  in each time slot, i.e.,  $\Phi_1(t) \geq \Phi_1^{*\nu}(t)$ ,  $\Phi_2(t) \geq \Phi_2^{*\nu}(t)$  and  $\Phi_4(t) \geq \Phi_4^{*\nu}(t)$ . Then, for time  $t = nT + \tau$  with  $\tau \in [0, T - s^{max}]$ , we have that

$$\begin{aligned} & B + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(t)) \epsilon_{mi} - \mathbb{E}(\Phi_1(t)) - \mathbb{E}(\Phi_2(t)) \\ & \quad - \mathbb{E}(\Phi_3(t)) - \mathbb{E}(\Phi_4(t)) \\ &\leq B + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(t)) \cdot \epsilon_{mi} - \mathbb{E}(\Phi_1^{*\nu}(t)) - \mathbb{E}(\Phi_2^{*\nu}(t)) + \tau B' \\ & \quad - (1 - \delta)(1 - \theta) \mathbb{E}(\Phi_3^{*0}(nT)) - \mathbb{E}(\Phi_4^{*\nu}(t)) \end{aligned}$$

For time  $t = nT + \tau$  with  $\tau \in (T - s^{max}, T)$ , we have that

$$\begin{aligned} & B + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(t)) \cdot \epsilon_{mi} - \mathbb{E}(\Phi_1(t)) - \mathbb{E}(\Phi_2(t)) \\ & \quad - \mathbb{E}(\Phi_3(t)) - \mathbb{E}(\Phi_4(t)) \\ &\leq B + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(t)) \cdot \epsilon_{mi} - \mathbb{E}(\Phi_1^{*\nu}(t)) - \mathbb{E}(\Phi_2^{*\nu}(t)) - \mathbb{E}(\Phi_4^{*\nu}(t)), \end{aligned}$$

which is based on the fact that  $\Phi_3(t) \geq 0$ .

Recalling the *drift-plus-penalty* inequality (15), we have that, for time  $t = nT + \tau$  with  $\tau \in [0, T - s^{max}]$ ,

$$\begin{aligned} & V \mathbb{E} \left( \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\eta_{mi}(t) \cdot s_m) - \beta d_{mi}(t) \cdot s_m) \right) \\ &\geq \mathbb{E}(\Phi_1^{*\nu}(t)) + \mathbb{E}(\Phi_2^{*\nu}(t)) - \tau B' + (1 - \delta)(1 - \theta) \mathbb{E}(\Phi_3^{*0}(nT)) \\ & \quad + \mathbb{E}(\Phi_4^{*\nu}(t)) - B - \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(t)) \cdot \epsilon_{mi}, \end{aligned}$$

for time  $t = nT + \tau$  with  $\tau \in (T - s^{max}, T)$ ,

$$\begin{aligned} & V\mathbb{E}\left(\sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\eta_{mi}(t) \cdot s_m) - \beta d_{mi}(t) \cdot s_m)\right) \\ & \geq \mathbb{E}(\Phi_1^{*\nu}(t)) + \mathbb{E}(\Phi_2^{*\nu}(t)) + \mathbb{E}(\Phi_4^{*\nu}(t)) - B \\ & \quad - \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(t)) \cdot \epsilon_{mi}. \end{aligned}$$

Summing up the above two inequalities for  $t = nT + \tau$  with  $\tau \in [0, T)$ , we have that

$$\begin{aligned} & V \sum_{\tau \in [0, T)} \mathbb{E}\left(\sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\eta_{mi}(nT + \tau) \cdot s_m) \right. \\ & \quad \left. - \beta d_{mi}(nT + \tau) \cdot s_m)\right) \\ & \geq \sum_{\tau \in [0, T)} (\mathbb{E}(\Phi_1^{*\nu}(t)) + \mathbb{E}(\Phi_2^{*\nu}(t)) + \mathbb{E}(\Phi_4^{*\nu}(t))) \\ & \quad + (1 - \delta)(1 - \theta)(T - s^{max} + 1)\mathbb{E}(\Phi_3^{*0}(nT)) \\ & \quad - B' \sum_{\tau \in [0, T - s^{max}]} \tau - TB \\ & \quad - \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(nT + \tau)) \cdot \epsilon_{mi}. \end{aligned}$$

We expand the right-hand side of the above inequality and have that

$$\begin{aligned} & \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} V\mathbb{E}(U(\eta_{mi}(nT + \tau) \cdot s_m) - \beta d_{mi}(nT + \tau) s_m) \\ & \geq \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} V\mathbb{E}(U(\eta_{mi}^{*\nu}(nT + \tau) \cdot s_m) - \beta d_{mi}^{*\nu}(nT + \tau) s_m) \\ & \quad + \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Y_{mi}(t) \cdot s_m \cdot (r_{mi}^{*\nu}(nT + \tau) - \eta_{mi}^{*\nu}(nT + \tau))) \\ & \quad - \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(r_{mi}^{*\nu}(nT + \tau) \cdot s_m \cdot Q_{mi}(nT + \tau)) \\ & \quad + (1 - \delta)(1 - \theta)(T - s^{max} + 1) \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(\mu_{mi}^{*0}(nT) \\ & \quad \times (Q_{mi}(nT) + Z_{mi}(nT))) \\ & \quad + \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(d_{mi}^{*\nu}(nT + \tau) s_m (Q_{mi}(nT + \tau) \\ & \quad + Z_{mi}(nT + \tau))) \\ & \quad - B' \frac{(T - s^{max} + 1)(T - s^{max})}{2} - TB \\ & \quad - \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(nT + \tau)) \cdot \epsilon_{mi} \\ & \geq \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} V(U(\eta_{mi}^{*\nu} \cdot s_m) - \beta d_{mi}^{*\nu} s_m) \\ & \quad - T \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(r_{mi}^{*\nu} \cdot s_m \cdot Q_{mi}(nT)) \\ & \quad - \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \tau (A_{mi}^{max} s_m)^2 \\ & \quad - T \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(Z_{mi}(nT) \cdot \epsilon_{mi}) \\ & \quad - \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \tau \epsilon_{mi}^2 \\ & \quad + (1 - \delta)(1 - \theta)(T - s^{max} + 1) \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(\mu_{mi}^{*0} \\ & \quad \times (Q_{mi}(nT) + Z_{mi}(nT))) \end{aligned}$$

$$\begin{aligned} & + \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} \mathbb{E}(d_{mi}^{*\nu} s_m (Q_{mi}(nT) + Z_{mi}(nT))) \\ & \quad - \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (2d_{mi}^{max} s_m \tau (1 + d_{mi}^{max} s_m)) \\ & \quad - B' \frac{(T - s^{max} + 1)(T - s^{max})}{2} - TB. \end{aligned}$$

The second inequality comes from the facts that:

- Since the stationary randomized algorithm should stabilize the network, each virtual queue  $Y_{mi}(t)$  is also stable. Thus, we have that

$$\mathbb{E}(r_{mi}^{*\nu}(nT + \tau) - \eta_{mi}^{*\nu}(nT + \tau)) \geq 0.$$

- According to the queueing law in Eqn. (2), we have that  $Q_{mi}(nT + \tau) - Q_{mi}(nT) \leq \tau A_{mi}^{max} s_m$ . Meanwhile, it is a fact that  $r_{mi}^{*\nu} = \mathbb{E}(r_{mi}^{*\nu}(nT + \tau))$ . So, we further have that

$$\begin{aligned} & \mathbb{E}(r_{mi}^{*\nu}(nT + \tau) \cdot s_m \cdot Q_{mi}(nT + \tau)) \\ & \leq \mathbb{E}(r_{mi}^{*\nu} \cdot s_m \cdot (Q_{mi}(nT) + \tau A_{mi}^{max} s_m)) \\ & \leq \mathbb{E}(r_{mi}^{*\nu} \cdot s_m \cdot Q_{mi}(nT) + \tau (A_{mi}^{max} s_m)^2). \end{aligned}$$

- According to the queueing law in Eqn. (12), we have that  $Z_{mi}(nT + \tau) - Z_{mi}(nT) \leq \tau \epsilon_{mi}$ . So, we further have that

$$\begin{aligned} & \mathbb{E}(\epsilon_{mi} \cdot Z_{mi}(nT + \tau)) \\ & \leq \mathbb{E}(\epsilon_{mi} \cdot (Z_{mi}(nT) + \tau \epsilon_{mi})) \\ & \leq \mathbb{E}(\epsilon_{mi} \cdot Z_{mi}(nT) + \tau (\epsilon_{mi})^2). \end{aligned}$$

- According to the queueing laws in Eqn. (2) and (12), we have that  $Q_{mi}(nT + \tau) - Q_{mi}(nT) \geq \tau(1 + d_{mi}^{max} s_m)$  and  $Z_{mi}(nT + \tau) - Z_{mi}(nT) \geq \tau(1 + d_{mi}^{max} s_m)$ . Based on its definition, we also know that  $d_{mi}^{*\nu} = \mathbb{E}(d_{mi}^{*\nu}(nT + \tau))$ . Thus, we have that

$$\begin{aligned} & \mathbb{E}(d_{mi}^{*\nu}(nT + \tau) \cdot s_m \cdot (Q_{mi}(nT + \tau) + Z_{mi}(nT + \tau))) \\ & \geq \mathbb{E}(d_{mi}^{*\nu} \cdot s_m \cdot (Q_{mi}(nT) + Z_{mi}(nT) - 2\tau(1 + d_{mi}^{max} s_m))) \\ & \geq \mathbb{E}(d_{mi}^{*\nu} \cdot s_m \cdot (Q_{mi}(nT) + Z_{mi}(nT)) \\ & \quad - 2d_{mi}^{max} \cdot s_m \tau (1 + d_{mi}^{max} s_m)). \end{aligned}$$

Let  $1 - \nu = \frac{(1 - \delta)(1 - \theta)(T - s^{max} + 1)}{T}$ . We have that  $\frac{(1 - \delta)(1 - \theta)(T - s^{max} + 1)}{T} \mu_{mi}^{*0} = \mu_{mi}^{*\nu}$ .

Since the stationary randomized algorithm stabilized the network, including all packet queues, we know that

$$\begin{aligned} r_{mi}^{*\nu} s_m & \leq \mu_{mi}^{*\nu} + d_{mi}^{*\nu} s_m, \\ \epsilon_{mi} & \leq \mu_{mi}^{*\nu} + d_{mi}^{*\nu} s_m. \end{aligned}$$

Then, we can have that

$$\begin{aligned} & \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} V\mathbb{E}(U(\eta_{mi}(nT + \tau) \cdot s_m) - \beta d_{mi}(nT + \tau) s_m) \\ & \geq \sum_{\tau \in [0, T)} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} V(U(\eta_{mi}^{*\nu} \cdot s_m) - \beta d_{mi}^{*\nu} s_m) \\ & \quad - \frac{T(T - 1)}{2} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} [(A_{mi}^{max} s_m)^2 + \epsilon_{mi}^2] \\ & \quad - \frac{T(T - 1)}{2} \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (2d_{mi}^{max} s_m (1 + d_{mi}^{max} s_m)) \\ & \quad - B' \frac{(T - s^{max} + 1)(T - s^{max})}{2} - TB. \end{aligned}$$



On both sides, we sum up over all  $n \geq 0$ , divide by  $VnT$  and take limits on  $n \rightarrow \infty^3$ . We can get that

$$\begin{aligned} & \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\bar{\eta}_{mi} \cdot s_m) - \beta \bar{d}_{mi} s_m) \\ \geq & \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\eta_{mi}^{*\nu} \cdot s_m) - \beta d_{mi}^{*\nu} s_m) \\ & - \frac{(T-1)}{2V} B'' - B' \frac{(T-s^{max}+1)(T-s^{max})}{2VT} - B/V, \end{aligned}$$

with  $B'' = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} ((A_{mi}^{max} s_m)^2 + \epsilon_{mi}^2 + 2d_{mi}^{max} s_m (1 + d_{mi}^{max} s_m))$ .

With Algorithm 1, we know that  $\bar{r}_{mi} \geq \bar{\eta}_{mi}$ . Meanwhile, the stationary randomized algorithm can make  $r^{*\nu} = \eta^{*\nu}$ . Then, we finally have that

$$\begin{aligned} & \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(\bar{r}_{mi} \cdot s_m) - \beta \bar{d}_{mi} s_m) \\ \geq & \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(r_{mi}^{*\nu} \cdot s_m) - \beta d_{mi}^{*\nu} s_m) \\ & - \frac{(T-1)}{2V} B'' - B' \frac{(T-s^{max}+1)(T-s^{max})}{2VT} - B/V, \end{aligned}$$

■

**Remark (throughput-delay tradeoff):** If we let  $V \rightarrow \infty$ , the queue lengths will grow to infinitely large (Lemma 1). With Lemma 1 and Theorem 3, we see that  $(1-\delta)(1-\theta) \rightarrow 1$  in this case. If we further let  $T \rightarrow \infty$ , we will have  $1-\nu = \frac{(1-\delta)(1-\theta)(T-s^{max}+1)}{T} \rightarrow 1$ , which means  $\sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{E}} (U(r_{mi}^{*\nu} \cdot s_m) - \beta d_{mi}^{*\nu} s_m)$  will be arbitrarily close to the offline optimum within the full capacity region instead of a fraction. In addition, if  $T/V \rightarrow 0$ , the constant utility gap will become  $\frac{(T-1)}{2V} B'' + B' \frac{(T-s^{max}+1)(T-s^{max})}{2VT} + B/V \rightarrow 0$ . Nevertheless, with Theorem 2, we have to be able to tolerate a long worst-case delay, which is proportional to  $V$ .

In conclusion, there is a tradeoff between the utility optimality and the tolerable worst-case delay. If  $V \rightarrow \infty$ ,  $T \rightarrow \infty$  and  $T/V \rightarrow 0$ , we will achieve an utility arbitrarily close to the offline optimum at the cost of infinitely large delay.

## VI. CONCLUSION AND REMARKS

In this paper, we investigate the optimal design of CSMA-based wireless communication to achieve a time-averaged maximum throughput utility with worst-case delay bounds and non-uniform job sizes, in general network settings. A two-timescale dynamic algorithm is proposed with dynamic decisions on rate control, link scheduling, job transmission and dropping in each time slot. Through rigorous analysis, we demonstrate that the proposed protocol can achieve a throughput utility arbitrarily close to its offline optima for jobs with non-uniform sizes, with a worst-case delay guarantees and a tradeoff of infinitely large maximum allowable delay. As our future work, we will explore the optimal CSMA protocol design in multi-channel settings with provable delay performance for general network topologies.

## REFERENCES

- [1] P.-K. Huang and X. Lin. Improving the delay performance of csma algorithms: A virtual multi-channel approach. In *Proc. of IEEE INFOCOM'13*, 2013.
- [2] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand. Fast mixing of parallel glauber dynamics and low-delay csma scheduling. In *Proc. of IEEE INFOCOM'11*, 2011.
- [3] L. Jiang, J. Ni, R. Srikant, and J. Walrand. Performance bounds of distributed csma scheduling. In *Proc. of ITA'10*, 2010.
- [4] L. Jiang and J. Walrand. Convergence analysis of a distributed csma algorithm for maximal throughput in a general class of networks. Technical report, UC Berkeley, 2008.
- [5] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *ACM/IEEE Transactions on Networking*, 18:960 – 972, 2010.
- [6] 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 IEEE ISIT'12*, 2012.
- [7] C.-H. Lee, D. Y. Eun, S.-Y. Yun, and Y. Yi. From glauber dynamics to metropolis algorithm: Smaller delay in optimal csma. In *Proc. of IEEE ISIT'12*, 2012.
- [8] B. Li and A. Eryilmaz. A fast-csma algorithm for deadline-constrained scheduling over wireless fading channels. In *Proc. of RAWNET'11*, 2011.
- [9] H. Li and N. Vaidya. Optimal csma-based wireless communication with worst-case delay and non-uniform sizes. Technical report, Available online: <http://arxiv.org/abs/1401.3511>, 2014.
- [10] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24:1452 – 1463, 2006.
- [11] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor. Maximizing utility via random access without message passing. Technical report, Microsoft Research, 2008.
- [12] M. Lotfinezhad and P. Marbach. Throughput-optimal random access with order-optimal delay. In *Proc. of IEEE INFOCOM'11*, 2011.
- [13] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan&Claypool Publishers, 2010.
- [14] M. J. Neely. Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks. In *Proc. of IEEE INFOCOM'11*, 2011.
- [15] 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. of IEEE INFOCOM'10*, 2010.
- [16] D. Shah and J. Shin. Delay optimal queue-based csma. In *Proc. of ACM SIGMETRICS'10*, 2010.
- [17] D. Shah, D. Tse, and J. N. Tsitsiklis. Hardness of low delay network scheduling. In *Proc. of Information Theory Workshop*, 2010.
- [18] V. Subramanian and M. Alanyali. Delay performance of csma in networks with bounded degree conflict graphs. In *Proc. of IEEE ISIT'11*, 2011.
- [19] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:1936 – 1948, 1992.

<sup>3</sup>It should be noted that  $\bar{a} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau \in [1, t]} \mathbb{E}(a(\tau))$