

Limiting Properties of Overloaded Multiuser Wireless Systems With Throughput-Optimal Scheduling

Xuan Wang, *Student Member, IEEE*, and Lin Cai, *Senior Member, IEEE*

Abstract—Throughput-optimal scheduling has been widely discussed due to its capability to stabilize single-hop multiuser wireless systems if possible. However, most of the previous discussions focused on the underloaded scenario, i.e., the arrival rate lies inside the achievable rate region. The behavior of throughput-optimal scheduling in overloaded multiuser wireless systems is the focus of this paper. We first show that, with the infinite buffer assumption, although all the queues are unstable, both the average throughput and a function of queue length converge as time evolves. In addition, the average throughput is the solution to a convex optimization problem whose objective is determined by the scheduling algorithm. By investigating the average throughput of two special throughput-optimal scheduling algorithms, i.e., the generalized MaxWeight and Log-Rule, we find that users can be prioritized by tuning the parameters associated with the scheduling algorithm, but the fairness is not likely to be guaranteed and some users may starve. Second, by studying the finite buffer system, we show that whether the buffer is dedicated to each queue or shared among queues has a great impact on the system performance, and the potential user starvation problem can be alleviated by a proper design.

Index Terms—Wireless networks, throughput-optimal scheduling, resource allocation, performance analysis.

I. INTRODUCTION

THROUGHPUT-OPTIMAL scheduling [8] is a class of important scheduling policies in multiuser wireless systems. It can explore the link quality variation as those utility-based scheduling policies discussed in [1], [2], and provide the maximal stability region in a network with stochastic traffic, which is superior to the utility-based scheduling whose stability region is smaller than the capacity region in general [3].

The first throughput-optimal scheduling algorithm was proposed two decades ago [4]. The authors studied the link scheduling problem in a centralized wireless system with ON-OFF channel and proposed a MaxWeight scheduling algorithm to stabilize the system. Later, different kinds of throughput-optimal scheduling algorithms were proposed to provide diverse features under various system assumptions. The

Manuscript received December 1, 2013; revised April 23, 2014, July 5, 2014, and August 27, 2014; accepted September 15, 2014. Date of publication September 23, 2014; date of current version October 17, 2014. The associate editor coordinating the review of this paper and approving it for publication was J. Widmer.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: xuan_wang@ece.uvic.ca; cai@ece.uvic.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCOMM.2014.2359875

probability of max-queue overflow is asymptotically minimized by the scheduling algorithms proposed in [5]–[7]. The queueing delay is minimized by the algorithms proposed in [8], [9]. Scheduling algorithms that can provide better delay performance compared with MaxWeight were proposed in [10], [11]. The system in the presence of heavy-tailed traffic can be stabilized by the algorithms proposed in [12], [13]. General guidelines about the necessary and sufficient conditions of the throughput-optimal scheduling were discussed in [14]–[17].

The performance of such throughput-optimal scheduling has been extensively investigated under the assumption that the system is stable, or underloaded. However, it is inevitable that a system may experience overloaded periods in practice due to the fluctuation of the traffic volume [18]. Therefore, it is important to characterize the system behavior in overloaded periods which has not received sufficient attention yet. The state-of-the-art research in this area has concluded only for some special throughput-optimal scheduling policies, such as MaxWeight scheduling in [24] and general-MaxWeight scheduling in [23]. The general system behavior of an overloaded system is still missing.

To fit the gap, in this work, we have studied the limiting properties of overloaded multiuser wireless systems with infinite buffer and a throughput-optimal scheduling policy similar to [15]. We have quantified the network performance of two throughput-optimal scheduling algorithms, the generalized MaxWeight (GMW) [15], [17] and the Log-Rule scheduling [8]. With the same throughput-optimal scheduling policy, we have further analyzed the performance of a finite-buffer system with Drop-Tail queue [27] and various buffer-sharing schemes, which is of practical interests and often missed in the literature.

We have made the following key observations: first, with infinite buffer, when the system is overloaded, all the queues in the system are unstable and the network converges to a fixed point formed by the average throughput and the scheduling function of queue length. Furthermore, the average throughput can be easily obtained by solving a convex optimization problem. Second, with the GMW algorithm, strict priority can be given to users by a proper parameter, so the QoS of users can be tuned easily. If all the users have the same priority, the system fairness in terms of Jain's index [26] cannot be guaranteed, but the scheduler can achieve certain fairness for the blocked/queued traffic which echoes the results in [24]. With the Log-Rule scheduling, the average throughput is identical to that with an asymptotic GMW scheduling. Third, if the system is subject to a shared buffer constraint, i.e., the buffer is shared among

queues, then the average throughput converges to a value which is not related to the type of the throughput-optimal scheduling algorithm. If each queue has its dedicated buffer, some users might suffer starvation, and some might achieve rate stability (average throughput equals its average arrival rate), depending on their buffer sizes.

The rest of the paper is organized as follows. Section II presents the related work. Section III introduces the system models and preliminaries, including the channel model, queueing model and scheduling policy. In Section IV, the limiting properties of the overloaded system are discussed, followed by the case study of the GMW and Log-Rule scheduling algorithms. The system performance with finite buffer is discussed in Section VI. Simulation results are given in Section VII, followed by the conclusion and further discussion.

In the following, bold face letter \mathbf{x} represents vector, x_i represents the i -th element in \mathbf{x} .

II. RELATED WORK

There are two categories of work discussing the system behavior in an overloaded network. One is to design a scheduling policy towards some specific objectives, such as providing a desired throughput in [19], aiming to stabilize part of the queueing system in [20], and dynamic routing to balance the overloaded traffic in [21]; the other is to discuss the system behavior under certain scheduling policies [22]–[25]. In [22], an overloaded network with α -fair scheduling was analyzed, and the asymptotic growth rate was obtained, which is a fixed-point of the system. In [23], the network behaviors of general-MaxWeight and α -fair scheduling policies in a multi-hop switched network were discussed. It showed that the queue size grows linearly with time for both scheduling policies, and the corresponding growth rates were characterized. In [24], the authors characterized the queue-size growth rate of the MaxWeight scheduling in parallel queues, and showed that the weight parameter can be tuned to achieve a certain fairness which is defined as a function of the growth rate.

Different from [22], [23], where a multi-hop network with fixed link service rate was discussed, we discuss a multiuser wireless system with time-varying service rate, similar to the network setting in [24]. On the other hand, the throughput-optimal scheduling discussed in our work is more general than that in [23] and [24], where (general)-MaxWeight scheduling policy only was discussed. Furthermore, we focus on the average throughput, while the previous work focused on the queue size of the overloaded system. Finally, we not only obtain the results for the infinite buffer case which is a similar assumption to the previous work, but also consider the finite buffer case, which is missed in the literature.

III. SYSTEM MODELS AND PRELIMINARIES

A. N -User Fading Broadcast Channel

We consider a system where N users communicate with a base station through a block fading broadcast channel. Within each time slot, the channel for each user is an additive white Gaussian noise (AWGN) one with a constant channel gain.

Across time slots, the channel gain for each user is independent and identically distributed. The user set is denoted by \mathcal{N} . For users in \mathcal{N} , the fading processes are independent of each other, jointly stationary and ergodic, but the statistical properties are not necessarily the same. Furthermore, there are mutual interference among users in any time instance. We further assume that the base station can obtain the channel state information (CSI) at the beginning of each time slot.

In time slot t , the achievable rate region is denoted by $\mathcal{C}(t)$, which is determined by the MAC layer and the physical layer protocols jointly. $\mathcal{C}(t)$ is always a convex and coordinate convex region, and no larger than the capacity region of the corresponding channel. If time division multiple access is used, $\mathcal{C}(t)$ is a convex and coordinate convex simplex; if superposition coding (SPC) and successive interference cancellation (SIC) is used, $\mathcal{C}(t)$ equals the capacity region of an N -user degraded Gaussian broadcast channel [37].¹ In addition, we assume that the maximal transmission rate for any user in time slot t is always positive, and therefore $\mathcal{C}(t)$ is always an N -dimensional region. By taking an average over time, we can obtain the average achievable rate region, which is the weighted Minkowski sum of $\mathcal{C}(t)$, denoted by \mathcal{C} , and is convex and coordinate convex.

B. Queueing Model

The network under consideration is a collection of FIFO queues. Data packets arrive randomly and are queued up in a buffer reserved for each user, and the arrival processes for different users are independent with each other. The resource is allocated at the beginning of each time slot based on the scheduling algorithm. Here, we first assume that the buffer size of each queue is infinity, and the finite buffer case will be discussed in Section VI. The state of the i -th buffer is the queue length and denoted by $q_i(t)$. Assume that the amount of allocated data to user i in time slot t is $r_i(t)$, whose vector form is $\mathbf{r}(t)$ and satisfies $\mathbf{r}(t) \geq \mathbf{0}$; the amount of arrived data in user i in time slot t is $a_i(t)$, whose vector form is $\mathbf{a}(t)$ and satisfies $\mathbf{a}(t) \geq \mathbf{0}$. All queue states form a vector $\mathbf{q}(t) \geq \mathbf{0}$, which is updated by: $\mathbf{q}(t+1) = [\mathbf{q}(t) - \mathbf{r}(t) + \mathbf{a}(t)]^+$, where $[\mathbf{x}]_i^+ = \max\{0, x_i\}$, $\forall i \in \mathcal{N}$. We further assume that $\{a_i(t), t = 1, 2, \dots\}$ is a sequence of independent and identically distributed random variables, and $a_i(1)$ has finite moments and satisfies $\lim_{A \rightarrow \infty} \sum_i^N A f_i(A) \Pr\{a_i(1) > A\} = 0$ where $f_i(\cdot)$ is the scheduling function to be explained later. This condition is used to guarantee that the tail of the arrival distribution decays fast enough compared to the scheduling function.² The average arrival rate is defined as $\lambda = \mathbb{E}[\mathbf{a}(1)]$, where \mathbb{E} is to take expectation.

¹Examples: Consider a two-user wireless system with a degraded Gaussian broadcast channel. The signal-to-noise-ratio (SNR) of user i is $\gamma_i(t)$, and assume $\gamma_1(t) > \gamma_2(t)$. If time division multiple access is used, then we have $\mathcal{C}(t) = \{(r_1, r_2) | r_1 / \log(1 + \gamma_1(t)) + r_2 / \log(1 + \gamma_2(t)) = 1\}$. If SPC and SIC are used, $\mathcal{C}(t) = \{(r_1, r_2) | r_1 = \log(1 + q\gamma_1(t)), r_2 = \log(1 + \gamma_2(t)) - \log(1 + q\gamma_2(t)), q \in [0, 1]\}$ [29], [30].

²The assumption is used to ensure that the scheduling policy (1) presented in the following subsection is throughput optimal. The analysis presented in this work does not rely on this assumption. More discussions on throughput-optimal scheduling in queueing networks with heavy-tailed traffic can refer to [38] and the reference therein.

C. Scheduling Policy

In this paper, we focus on the throughput-optimal scheduling policy proposed in [15], as it is one of the most general scheduling policy. To simplify our analysis, we slightly modify the original scheduling policy as shown below. The rate allocated to users in time slot t is based on the solution to the following weighted-sum-rate-maximization problem:

$$\mathbf{r}(t) \in \arg \max_{\boldsymbol{\eta} \in \mathcal{C}(t)} \sum_i f_i(q_i(t)) \eta_i, \quad (1)$$

and the ties are broken randomly, where η_i is the possible transmission rate of user i in slot t , and $\boldsymbol{\eta}$ is the possible transmission rate vector lies inside the instantaneous achievable rate region $\mathcal{C}(t)$, $f_i(x)$ is the scheduling function with $x \geq 0$ and satisfies the following conditions:

- 1) $\forall i$, $f_i(x)$ is a non-negative strictly increasing continuous function with $\lim_{x \rightarrow \infty} f_i(x) = \infty$.
- 2) Given any $C_1, C_2 > 0$ and $0 < \sigma < 1$, there is some $M > 0$ such that for all $x > M$, $\forall i$, $f_i(x)$ satisfies

$$(1 - \sigma)f_i(x) \leq f_i(x - C_1) \leq f_i(x + C_2) \leq (1 + \sigma)f_i(x).$$

Examples of function $f_i(x)$ include: $f_i(x) = bx^a$, $a > 0, b > 0$; $f_i(x) = b \log(x + a)$, $a > 0, b > 0$; $f_i(x) = e^{bx^a}$, $0 < a < 1, b > 0$. Note that $f_i(x) = e^{bx^a}$, $a \geq 1, b > 0$ does not satisfy condition 2).

Let $\bar{\mathbf{f}}(\mathbf{x})$ be the normalized weight in vector form, whose i -th component is denoted by $\bar{f}_i(\mathbf{x}) = \frac{f_i(x_i)}{\sum_i f_i(x_i)}$. Equivalently, the optimization problem (1) can be represented as follows:

$$\mathbf{r}(t) \in \arg \max_{\boldsymbol{\eta} \in \mathcal{C}(t)} \sum_i \bar{f}_i(\mathbf{q}(t)) \eta_i. \quad (2)$$

D. Stability

We adopt the definitions of stability presented in [28], which are shown as follows.

Definition 1: A queue q is weakly stable if, for every $\epsilon > 0$, there exists $B > 0$ such that $\limsup_{t \rightarrow \infty} \Pr\{q(t) > B\} < \epsilon$, where $q(t)$ is the queue length in time t .

Definition 2: A system of queues \mathbf{q} is weakly stable if, for every $\epsilon > 0$, there exists $B > 0$ such that $\limsup_{t \rightarrow \infty} \Pr\{\|\mathbf{q}(t)\| > B\} < \epsilon$, where $\|\mathbf{q}(t)\|$ is the Euclidean norm of $\mathbf{q}(t)$.

From the definition we can conclude that, if q is unstable, then for any $B > 0$, we have $\limsup_{t \rightarrow \infty} \Pr\{q(t) < B\} < \epsilon$, where $\epsilon > 0$ and is arbitrarily small; if q is stable, we can find a B such that for all the t , $\Pr\{q(t) < B\} > 1 - \epsilon$, where $\epsilon > 0$ and is arbitrarily small. If a system of queues is unstable, we can conclude that at least one queue is unstable.

We further define the overloaded system as follows.

Definition 3: An overloaded system is defined as a system such that the average arrival rate vector $\boldsymbol{\lambda}$ lies outside the average achievable rate region \mathcal{C} , i.e., $\boldsymbol{\lambda} \notin \mathcal{C}$.

IV. LIMITING PROPERTIES

As shown in [15], with scheduling policy (1), if the queueing system is an aperiodic Markov chain and the mean arrival rate lies inside the achievable rate region, then the Markov chain is positive recurrent, or the system of queues is weakly stable. Whether the system is able to be strongly stable further depends on function f_i . As indicated in [28], weak stability implies that the offered load can be processed by the server, but the delay performance cannot be guaranteed. Consequently, if the system is overloaded, the system is unable to be weakly stable. Without confusion, stable means weakly stable, and unstable means unable to be weakly stable in the following.

A. Stability Property

From the definition of weak stability, it is unclear whether all the individual queues in the system are unstable, or only part of the queues in the system are unstable. We have Theorem 1 to answer this question.

Theorem 1: Given infinite buffer, for a multiuser wireless system with throughput-optimal scheduling as (1), if the system is overloaded, then all the queues are unstable.

Proof: See Appendix A. ■

As all the queues are unstable, then based on the stability properties shown in Section III-D, when $t \rightarrow \infty$, $\mathbf{q}(t) - \mathbf{r}(t) + \mathbf{a}(t) > \mathbf{0}$ holds with probability $1 - \epsilon$, where ϵ is arbitrarily small, which suggests that the allocated rate will not be wasted due to the shortage of data in the queue. Therefore, when obtaining the average throughput, we can use the allocated rate instead of the transmitted data size in each slot.

B. Average Throughput and Fixed Point of the System

Theorem 2: For an overloaded multiuser wireless system with scheduling and resource allocation algorithm as in (1), the corresponding average throughput of users in the system converges, i.e., for any t_0 ,

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{r}(t_0 + t) \rightarrow \boldsymbol{\mu}.$$

$\boldsymbol{\mu}$ is a solution to the following problem

$$\max_{\boldsymbol{\eta} \in \mathcal{C}^\lambda} \sum_i \lim_{t \rightarrow \infty} \bar{f}_i((\boldsymbol{\lambda} - \boldsymbol{\mu})t + \boldsymbol{\mu}'(t)) \bar{\eta}_i, \quad (3)$$

where $\bar{\eta}_i$ is the variable in the above optimization problem and represents the average throughput of user i , and $\bar{\eta}$ is that in the vector form, $\mathcal{C}^\lambda = \{\bar{\eta} | \bar{\eta} \in \mathcal{C}, \bar{\eta} \leq \boldsymbol{\lambda}\}$ is the constrained average achievable rate region, as the average throughput should be no larger than the average arrival rate, and $\boldsymbol{\mu}'(t)$ is an auxiliary variable with $\lim_{t \rightarrow \infty} \boldsymbol{\mu}'(t)/t = \mathbf{0}$. Furthermore, $\boldsymbol{\mu}$ is unique, and is the solution to the following problem

$$\lim_{t \rightarrow \infty} \max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i F_i((\lambda_i - \bar{\eta}_i)t), \quad (4)$$

where $F_i(x)$ is an antiderivative of $f_i(x)$.

Proof: See Appendix B. ■

From Theorem 2, although the system is overloaded, the average throughput exists and converges to μ which is a solution to (3). From (3), it is noted that μ cannot be directly obtained. Also, by observing (3), we can conclude that the average throughput and the scheduling function of queue length form a unique fixed point of the system.

Since the average throughput of the system converges to the solution of (4), based on the above approach, we can obtain the average throughput by giving the detailed system assumptions. Note that in [24], the author obtained a similar result for the MaxWeight scheduling, which is a special case of (4). However, the result obtained in [24] cannot be extended to a system with a general throughput-optimal scheduling algorithm, as the result relies on the linear structure of function $f_i(x)$ in the MaxWeight scheduling.

The scheduling algorithm according to (1) is an online algorithm to solve (4). Since $F_i((\lambda_i - \bar{\eta}_i)t)$ is a function of the average arrival rate, and $\bar{\eta}_i$ is linearly impacted by λ_i , we can conclude that the average throughput is also related to the arrival rate in general. This suggests that, in the overloaded system with scheduling algorithm as (1), if Jain's index (which is a function of the average throughput) is used to quantify the system fairness, then it is likely that such a fairness index is not only related to the throughput, but also impacted by the arrival rate. This fairness issue will be elaborated in the following section, by studying two sample scheduling algorithms.

V. EXAMPLES: THE GMW AND LOG-RULE SCHEDULING ALGORITHMS

In this section, we study two representative throughput-optimal scheduling algorithms, the GMW [15], [17] and the Log-Rule [8]. We first discuss how to solve the optimization problem to obtain the average throughput and the impact of the parameters on the average throughput, followed by the fairness issue.

A. Generalized MaxWeight

For the GMW, we have $f_i(x) = b_i x^{\alpha_i}$, where $b_i > 0$ and $\alpha_i > 0$. Then,

$$F_i((\lambda_i - \bar{\eta}_i)t) = -\frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}t^{\alpha_i},$$

and the average throughput is the solution to the following problem

$$\lim_{t \rightarrow \infty} \min_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i \frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}t^{\alpha_i}. \quad (5)$$

Easily we can see that α_i is critical to solve the problem, and the user with the largest α_i dominates the objective of (5). Therefore, we can adopt an iterative greedy approach to solve (5) as follows.

First we divide the user set \mathcal{N} into K groups $\{\mathcal{G}_k\}$, $k = 1, 2, \dots, K$. The users in the same group have the same α_i ,

i.e. $\forall i \in \mathcal{G}_k$, $\alpha_i = \alpha_{\mathcal{G}_k}$. The groups are ordered decreasingly according to α_i . i.e., if $m < n$, then $\alpha_{\mathcal{G}_m} > \alpha_{\mathcal{G}_n}$.

Suppose that the average throughput of user $i \in \bigcup_{m=1}^{k-1} \mathcal{G}_m$ is $\bar{\eta}_i^*$. Then the solution to problem (5) is the solution to the following problem

$$\begin{aligned} \lim_{t \rightarrow \infty} \min & \sum_i \frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}t^{\alpha_i - \alpha_{\mathcal{G}_k}}, \\ \text{s.t. } & \bar{\eta} \in \mathcal{C}^\lambda, \forall i \in \bigcup_{m=1}^{k-1} \mathcal{G}_m, \bar{\eta}_i = \bar{\eta}_i^*, \end{aligned}$$

which is further equivalent to

$$\begin{aligned} \lim_{t \rightarrow \infty} \min & \sum_{i \in \bigcup_{j=1}^{k-1} \mathcal{G}_j} \frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}t^{\alpha_i - \alpha_{\mathcal{G}_k}} \\ & + \sum_{i \in \mathcal{G}_k} \frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}, \\ \text{s.t. } & \bar{\eta} \in \mathcal{C}^\lambda, \forall i \in \bigcup_{m=1}^{k-1} \mathcal{G}_m, \bar{\eta}_i = \bar{\eta}_i^*. \end{aligned}$$

For the obtained average throughput $\bar{\eta}_i^*$, we have either $\forall i \in \bigcup_{j=1}^{k-1} \mathcal{G}_j$, $\bar{\eta}_i^* = \lambda_i$, or $\exists i \in \bigcup_{j=1}^{k-1} \mathcal{G}_j$, $\bar{\eta}_i^* \neq \lambda_i$. For the first case, the solution to (5) is the solution to the following problem

$$\min \sum_{i \in \mathcal{G}_k} \frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}, \quad (6a)$$

$$\text{s.t. } \bar{\eta} \in \mathcal{C}^\lambda, \forall i \in \bigcup_{m=1}^{k-1} \mathcal{G}_m, \bar{\eta}_i = \bar{\eta}_i^*. \quad (6b)$$

For the second case, it implies that $\bar{\eta}_i^* < \lambda_i$, which further implies that $\bar{\eta}^*$ lies on the boundary of \mathcal{C}^λ , where $\bar{\eta}^* = \{\bar{\eta}_i : \bar{\eta}_i = \bar{\eta}_i^*, \text{ if } i \in \bigcup_{j=k+1}^K \mathcal{G}_j; \bar{\eta}_i = 0, \text{ if other wise}\}$. By summarizing the above two cases, we can conclude that the solution to (5) is also the solution to the following problem

$$\begin{aligned} \min & \sum_{i \in \bigcup_{j=1}^k \mathcal{G}_j} \frac{b_i}{\alpha_i + 1}(\lambda_i - \bar{\eta}_i)^{\alpha_i+1}, \\ \text{s.t. } & \bar{\eta} \in \mathcal{C}^\lambda, \forall i \in \bigcup_{m=1}^{k-1} \mathcal{G}_m, \bar{\eta}_i = \bar{\eta}_i^*, \end{aligned}$$

which further implies that the average throughputs of users in group \mathcal{G}_k can be obtained by solving the above problem, as the solution is unique.

Iteratively, the average throughput is obtained.

1) *Impact of Scheduling Parameters:* By observing the algorithm structure to obtain the average throughput, we can see that α_i and b_i are important to the performance of users. Note that in a stable system, α_i can be used to control the priority of the queue and improve the delay performance [7]. This priority only affects the delay performance, and it does not change the average throughput which equals the average arrival rate in an underloaded system. But in an overloaded system, since the scheduler allocates the available resource to the users in the decreasing order of α_i , a user with a larger α_i has a 'hard' higher priority. Therefore by increasing α_i of a user to a proper value (for instance, larger than all the other α_j , where $j \neq i$),

its average throughput can be improved. This behavior suggests that the QoS of a user can be improved by assigning a larger α_i . In summary, if the system is stable, then a larger α_i can result in a smaller delay; if the system is unstable, then it can result in a higher throughput.

Similar to α_i , b_i can also be used to differentiate the users, but within a group of users with the identical α_i . Note that the throughput of users in \mathcal{G}_k is either all zero, or can be obtained from problem (6). As $\forall i \in \mathcal{G}_k$, α_i are identical. Fixing b_j where $j \neq i$, r_i is possible to be increased w.r.t. b_i . A larger b_i generally means a possible larger average throughput, but it cannot guarantee the user gets served first. Therefore we consider the priority associated with b_i as a ‘soft’ priority.

2) *Fairness*: Since α_i is used to control the priority of user, and the incoming traffic is served strictly according to the priority, the fairness should only be considered within each group. Considering the scheduling algorithm with $\alpha_i = \alpha$, the average throughput is the solution to the following problem

$$\min_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i b_i (\lambda_i - \bar{\eta}_i)^{\alpha+1}, \quad (7)$$

which is a $L_{\alpha+1}$ -Norm minimization problem in a scaled space with constraint $\bar{\eta} \in \mathcal{C}^\lambda$, and $b_i^{1/\alpha}$ is the scale factor in dimension i .

Suppose that the scale factor for each dimension is identical, i.e., $\forall i$, $b_i = 1$, and then geometrically, the average throughput is a point in the constraint set \mathcal{C}^λ and has the minimal $L_{\alpha+1}$ distance to the point λ . Since \mathcal{C} and $\{\bar{\eta} | \bar{\eta} \leq \lambda\}$ are both convex, the constraint set \mathcal{C}^λ is also convex, so we can conclude that, as long as λ is not scaled proportional to $\lambda - \mu(\lambda)$, where $\mu(\lambda)$ is the solution for the given λ , the average throughput will change based on the change of λ . As a result, the fairness (in terms of Jain’s index) only makes sense for a given λ . For a system where λ is not under control, fairness cannot be guaranteed as any user can change the Jain’s index by increasing the arrival rate.

Although the fairness of the throughput cannot be guaranteed, the scheduling algorithm actually guarantees the fairness of the queued/blocked traffic. This can be seen from two asymptotic cases easily. When $\alpha \rightarrow 0$, the problem approaches $\max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i \bar{\eta}_i$ which means the user with a larger possible transmission rate will be satisfied first and users with smaller possible transmission rates may starve. Such a greedy allocation can also be interpreted as that the scheduling algorithm does not consider the fairness at all. When $\alpha \rightarrow \infty$ the problem approaches $\min_{\bar{\eta} \in \mathcal{C}^\lambda} |\lambda - \bar{\eta}|_\infty$, i.e., minimizing the L_∞ norm of vector $\lambda - \bar{\eta}$, which means the queued traffic satisfies a min-max fairness [34], the dual of the max-min fairness [32], [33]. From the above two asymptotic cases we can conclude that, the fairness of the blocked traffic can be guaranteed by choosing a proper α .

There are some other works discussed the fairness issue in some systems. In [24], the author proposed a fairness metric using the backlog growth direction in parallel queues, which is identical to the fairness of the queued/blocked traffic. Furthermore, the author showed that, for a specific backlog growth direction, by designing a proper b_i , the backlogged traffic is also

minimized. In [34], the author discussed how to minimizing an α -penalty function of the queue overflow rate in a server farm. It turns out that by choosing $\forall i$, $b_i = 1$, the problem (7) is a special construct of the problem discussed in [34]. However, as [24] and [34] both using identical α_i , i.e., only discussed the fairness within each group, their works are unable to find the critical impact of α_i on the fairness.

B. Log-Rule

By a slight modification to the original policy presented in [8], we have the equivalent Log-Rule which has $f_i(x) = b_i \log(1 + a_i x)$ with $a_i > 0$ and $b_i > 0$. Then

$$F_i((\lambda_i - \bar{\eta}_i)t) = - \left(\frac{b_i}{a_i t} + b_i(\lambda_i - \bar{\eta}_i) \right) \log(1 + a_i(\lambda_i - \bar{\eta}_i)t) + \frac{b_i}{a_i t} + (\lambda_i - \bar{\eta}_i)b_i,$$

and the average throughput is the solution to the following problem

$$\lim_{t \rightarrow \infty} \min_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i \left(\frac{b_i}{a_i t} + b_i(\lambda_i - \bar{\eta}_i) \right) \log(1 + a_i(\lambda_i - \bar{\eta}_i)t) - \frac{b_i}{a_i t} - (\lambda_i - \bar{\eta}_i)b_i,$$

which is equivalent to

$$\lim_{t \rightarrow \infty} \min_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i b_i(\lambda_i - \bar{\eta}_i) \log(1 + a_i(\lambda_i - \bar{\eta}_i)t), \quad (8)$$

by ignoring the terms that do not increase with t as they have no impact on the solution.

Note that the solution to (8) is also the solution to the following problem

$$\lim_{t \rightarrow \infty} \min_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i b_i(\lambda_i - \bar{\eta}_i) \log(1 + a_i(\lambda_i - \bar{\eta}_i)t) / \log(1 + t),$$

which is further equivalent to

$$\max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i b_i \bar{\eta}_i, \quad (9)$$

since

$$\lim_{t \rightarrow \infty} \log(1 + a_i(\lambda_i - \bar{\eta}_i)t) / \log(1 + t) = 1.$$

Problem (9) and problem (8) are not equivalent. But if (9) has a unique solution, then it is also the solution to (8).

Note that (9) is not related to a_i , the average throughput of the Log-Rule is only affected by parameter b_i . Comparing (9) to the GMW with $\alpha \rightarrow 0$, both schedulers have the same average throughput. Consequently, the discussions on the impact of b_i on the average throughput and the fairness issue are identical to the GMW with $\alpha \rightarrow 0$ case.

VI. PERFORMANCE IN A FINITE BUFFER SYSTEM

In the previous sections, we have discussed the limiting properties of the overloaded system with throughput-optimal scheduling. All the queues in the system are unstable, and each queue length increases to infinity. While with a more practical assumption that the buffer for the queue should be finite, the overloaded packets will be dropped by the queue management scheme. In this section, we give a discussion on the system performance in a finite buffer system.

A. System Assumption

Since the system is a collection of queues, all the queues can either share the same buffer, such as the downlink case of a wireless communication system, or each queue has its own dedicated buffer, such as the uplink of a wireless communication system. We further assume that the system uses the Drop-Tail scheme as the queue management scheme, and the arrival traffic for i -th flow (for user i) is a Poisson traffic with average arrival rate λ_i .

B. Shared Buffer Case

As the queue has finite buffer, the incoming packet will be dropped if it encounters the event that the buffer is full. The packet drop in a queueing system with a shared buffer is identical to that in a single queue with buffer size B_i^{\max} and the aggregated arrival traffic.

As the incoming traffic of each flow is Poisson traffic, the aggregated traffic is also Poisson. According to the Poisson Arrival See Time Average (PASTA) property, each packet encounters the event that the buffer is full with the same probability. So the packet dropping probability for each flow is identical and denoted as k^* . Consequently the packet dropping rate d is proportional to the packet arrival rate λ , and we have $d = k^* \lambda$. For the average throughput μ , we have $\mu = \lambda - d$. As the system is overloaded and throughput-optimal scheduling is used, μ should lie on the boundary of the capacity region \mathcal{C} , i.e., $\mu \in \text{bd}(\mathcal{C})$. Consequently, we have $\mu = (1 - k^*)\lambda$, and k^* is obtained from $k^* = \arg \min_{(1-k)\lambda \in \mathcal{C}} k$.

The average throughput is not affected by the type of the throughput-optimal scheduling algorithm, and is determined by the statistical properties of the arrival traffic and the queue management scheme. Furthermore, the average throughput is proportional to the average arrival rate, which is different from the infinite buffer case where some users may starve (such as GMW with heterogeneous α_i). In other words, for an overloaded system with finite shared buffer, the long-term (permanent) fairness may be improved, despite the fairness may be poor during the transient period (as the performance during the transient period is similar to the infinite buffer case), such as in a system using the GMW scheduling with heterogeneous α_i as discussed in Section V-A.

Note that as long as the packet dropping probability is identical for different flows, the above argument holds. Even though in a system with Drop-Tail scheme and bursty arrival traffic, the above property may not hold in general, however,

certain active queue management (AQM) scheme can be used, such as Random Early Detection [27], to retain this property.

C. Dedicated Buffer Case

As discussed in [35], the system can be modeled as a controlled random walk and can be further approximated by a deterministic fluid model, where the data packets for each user are modeled as a continuous fluid flow that enter and leave the buffer [36]. Each flow has its dedicated buffer with size B_i^{\max} , then all the B_i^{\max} will jointly determine the average throughput and therefore fairness. The corresponding fluid scheduling model is represented as $\mathbf{q}(t) = \mathbf{q}(0) - \mathbf{z}(t) + \boldsymbol{\lambda}t$, where $\mathbf{z}(t)$ is the cumulative allocated fluid resource up to time t .

As the system is overloaded and the fluid model is used to approximate the system, the system cannot be idle at the time it loses fluid. Therefore, the average throughput should be equal to the average allocated rate, i.e., $\mu = \lim_{t \rightarrow \infty} \mathbf{z}(t)/t$. We have $\mathbf{z}(t) = \boldsymbol{\mu}t + \boldsymbol{\mu}'(t)$, where $\lim_{t \rightarrow \infty} \boldsymbol{\mu}'(t)/t = 0$. Consequently, we have

$$\mathbf{q}(t) = \mathbf{q}(0) - \boldsymbol{\mu}t - \boldsymbol{\mu}'(t) + \boldsymbol{\lambda}t. \quad (10)$$

Suppose that the queues in set \mathcal{S} can achieve rate stability, i.e., the average throughput equals the average arrival rate, and the queues not in \mathcal{S} cannot achieve rate stability. We have $\forall i \in \mathcal{S}$, $\mu_i = \lambda_i$ and $\forall i \notin \mathcal{S}$, $\mu_i < \lambda_i$. Therefore, based on (10), $\forall i \notin \mathcal{S}$ there exists a t_b such that for all $t > t_b$, $f_i(q_i(t)) = f_i(B_i^{\max})$, i.e., the buffer of queue i is full after it has been filled up. Consequently the allocated rate in t is based on the following problem

$$\max_{\eta \in \mathcal{C}(t)} \sum_{i \notin \mathcal{S}} f_i(B_i^{\max}) \eta_i + \sum_{i \in \mathcal{S}} f_i(q_i(t)) \eta_i,$$

with random tie-breaking. By taking an average over time, we have $\boldsymbol{\mu}$ to be the solution to the following problem

$$\begin{aligned} \max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_{i \notin \mathcal{S}} f_i(B_i^{\max}) \bar{\eta}_i \\ + \sum_{i \in \mathcal{S}} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=t_b+1}^{t_b+T} f_i(q_i(t)) \bar{\eta}_i. \end{aligned} \quad (11)$$

Since $\forall i \in \mathcal{S}$, $\mu_i = \lambda_i$ and $f_i(q_i(t)) \leq f_i(B_i^{\max})$, substituting $f_i(B_i^{\max})$ for $f_i(q_i(t))$ does not change the solution. Thus, problem (11) is identical to the following problem

$$\max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i f_i(B_i^{\max}) \bar{\eta}_i, \quad (12)$$

with uniform tie-breaking.

In the above analysis, the key argument is that $\forall i \notin \mathcal{S}$ and $t > t_b$, $f_i(q_i(t)) = f_i(B_i^{\max})$. Therefore, for any arrival traffic as long as the above condition can approximately hold, the average throughput is close to the solution to (12).

Note that if the average arrival rate λ is sufficiently large, the constraint set \mathcal{C}^λ will be equal to the capacity region \mathcal{C} . Based on (12), the average throughput is no longer related to λ . Consequently, the system fairness in terms of Jain's index can be guaranteed by a properly designed buffer size B_i^{\max} .

With the finite buffer assumption, the system behavior is quite different from that with the infinite buffer assumption, which is typically used in the literature. The user starvation problem can be alleviated by the shared buffer scheme, while certain queues may achieve rate stability if the buffer size is set properly in the dedicated buffer case and we will demonstrate it with an example in Section VII-A2.

VII. PERFORMANCE EVALUATION

In this section, we validate our analytical results and compare the system performance based on different throughput-optimal scheduling algorithms and different system assumptions. During the evaluation, Poisson arrival traffic is used if not specified. For each simulation setting, we repeat the simulation multiple runs and the results demonstrate that the average throughput can converge to the theoretical value. Then, we take one run as a sample-path of the system to plot the results in the figures. The average throughput in time slot t is calculated by taking the average over the results of the previous 2000 time slots.

A. Two-User Static Channel Case

First considering a two-user static Gaussian broadcast channel (GBC), the signal-to-noise ratio of user i is γ_i , and assume $\gamma_1 > \gamma_2$. Then the achievable rate region is [37]

$$\mathcal{C} = \{\mathbf{r} | r_1 = \log_2(1 + q\gamma_1),$$

$$r_2 = \log_2(1 + \gamma_2) - \log_2(1 + q\gamma_2), 0 \leq q \leq 1\}.$$

This channel is a special case of the general N -user fading broadcast channel, since the stochastic process governing the transition of the channel state is deterministic. Such channel is discussed because of the strictly convex property of the resultant achievable rate region, and therefore the average throughputs for different simulation settings are always unique. Also by using the two-user GBC channel first and then the Markov channel in Section VII-B, we are able to demonstrate that the convergence of $f_i(\mathbf{q}(t))$ and average throughput does not depend on whether the channel is stochastic or not.

During the evaluation, we set $\gamma_1 = 100$ and $\gamma_2 = 10$.

1) *Infinite Buffer Case*: We first validate that with the GMW or Log-Rule scheduling, the average throughput and $\bar{f}(\mathbf{q}(t))$ converge. The results are shown in Figs. 1–3. In the figures illustrating the average throughput, the solid curve represents the throughput of user 1 and the dashed curve represents that of user 2. The analytical results are shown by points “ \times ” in all figures.

Fig. 1 illustrates the system behavior with the GMW scheduler and identical α . The average throughputs and $f_1(q_1)/f_2(q_2)$ quickly converge, and different b results in different converged value. Fixing $b_1 = 1$ and increasing b_2 from 1 to 10, the average throughput of user 2 also increases, but the network does not give a ‘hard’ priority to user 2. From the trend of throughput changes, with the further increasing of b_2 , the average throughput of user 2 will increase to the same as its arrival rate. Since $f_1(q_1)/f_2(q_2)$ is no more informative than the average throughput and the convergence of average throughput

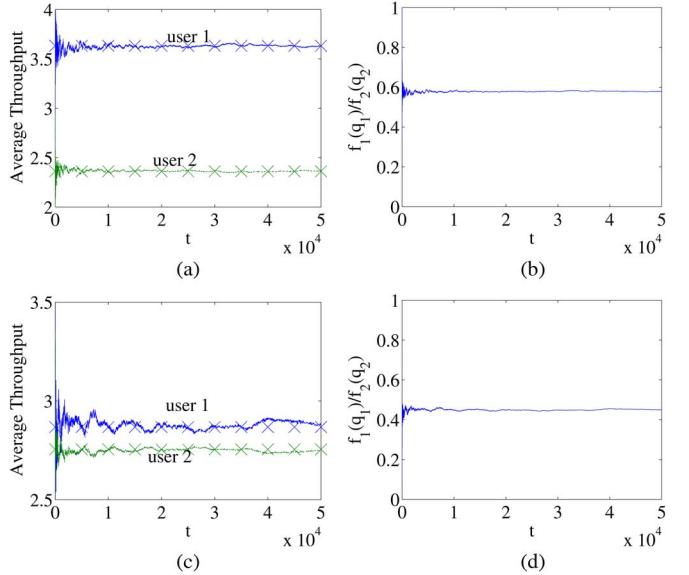


Fig. 1. The convergence of the average throughput and $\bar{f}(\mathbf{q}(t))$ of an infinite buffer network with GMW scheduler, $\alpha = [1 1]$, $\lambda = [4 3]$. (a) $\mathbf{b} = [1 1]$; (b) $\mathbf{b} = [1 1]$; (c) $\mathbf{b} = [1 10]$; (d) $\mathbf{b} = [1 10]$.

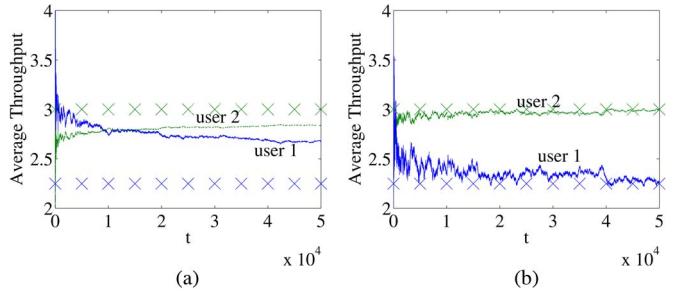


Fig. 2. The average throughput of an infinite buffer network with GMW scheduler, $\mathbf{b} = [1 1]$, $\lambda = [4 3]$. (a) $\alpha = [1 1.3]$; (b) $\alpha = [1 1.6]$.

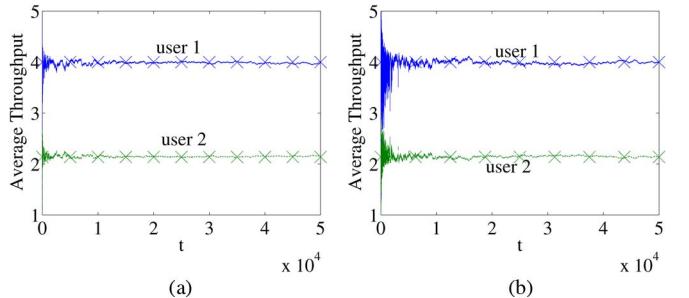


Fig. 3. The average throughput of an infinite buffer network, comparing Log-Rule scheduler with asymptotic GMW scheduler, $\mathbf{b} = [1 1]$, $\lambda = [4 3]$. (a) Log-Rule, $\alpha = [1 1]$; (b) GMW, $\alpha = [0.1 0.1]$.

indicates the convergence of $f_1(q_1)/f_2(q_2)$, in the following we only show the comparison of the average throughput.

Fig. 2 shows the behavior of a system with GMW scheduler and different α . With $\alpha = [1 1.3]$, the average throughput converges slowly, and cannot converge to the analytical results within 50 000 time slots. Changing α to [1 1.6], the system converges much fast, and the average throughput of user 2 converges to its arrival rate, which suggests that user 2 has a strictly higher priority compared with user 1.

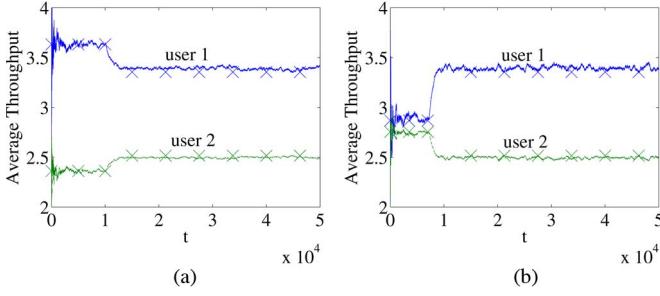


Fig. 4. The average throughput of a finite shared buffer network with GMW scheduler and Drop-Tail scheme, $\alpha = [1 \ 1]$, $\lambda = [4 \ 3]$, $B^{\max} = 10^4$. (a) $b = [1 \ 1]$; (b) $b = [1 \ 10]$.

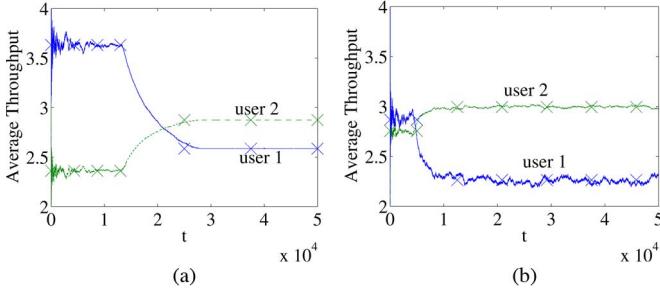


Fig. 5. The average throughput of a finite dedicated buffer network with GMW scheduler and Drop-Tail scheme, $\alpha = [1 \ 1]$, $\lambda = [4 \ 3]$, $B^{\max} = [5000 \ 12\,500]$. (a) $b = [1 \ 1]$; (b) $b = [1 \ 10]$.

We compare the Log-Rule with the asymptotic GMW in Fig. 3. For the asymptotic GMW, we choose $\alpha = 0.1$. The average throughputs of GMW and Log-Rule are identical and equal to the analytical results. Also we observe that the average throughput of user 1 equals its arrival rate, which is a result that the Log-Rule or the asymptotic GMW degrades to an algorithm which schedules the user with a larger channel rate first.

2) Finite Buffer Case: The system behavior under the finite buffer assumption is presented here. In order to observe the transient network behavior, we set the buffer size to be a relatively large value in different scenarios.

First we show the results of the shared buffer with the Drop-Tail queue scheme case. Fig. 4 shows the results of a network with the GMW scheduler with different b . As shown in the figure, the average throughput first converges to a transient value which is determined by the infinite buffer case, and thereafter converges to a permanent value. The transient value is determined by the parameters of GMW (parameter b), while the permanent fixed value is identical and independent of these parameters.

Similar to the shared buffer with the Drop-Tail queue scheme case, in the dedicated buffer with the Drop-Tail queue scheme case, the network first converges to a transient value, then converges to the permanent value, which is illustrated in Fig. 5. By changing the parameter b , not only the transient value changes, but also the permanent value changes. Since the permanent throughput of user 2 equals its average arrival rate, user 2 achieves rate stability.

B. Markov Channel Model

We further use a Markov channel model to illustrate the system dynamics with temporarily overloaded arrival traffic.

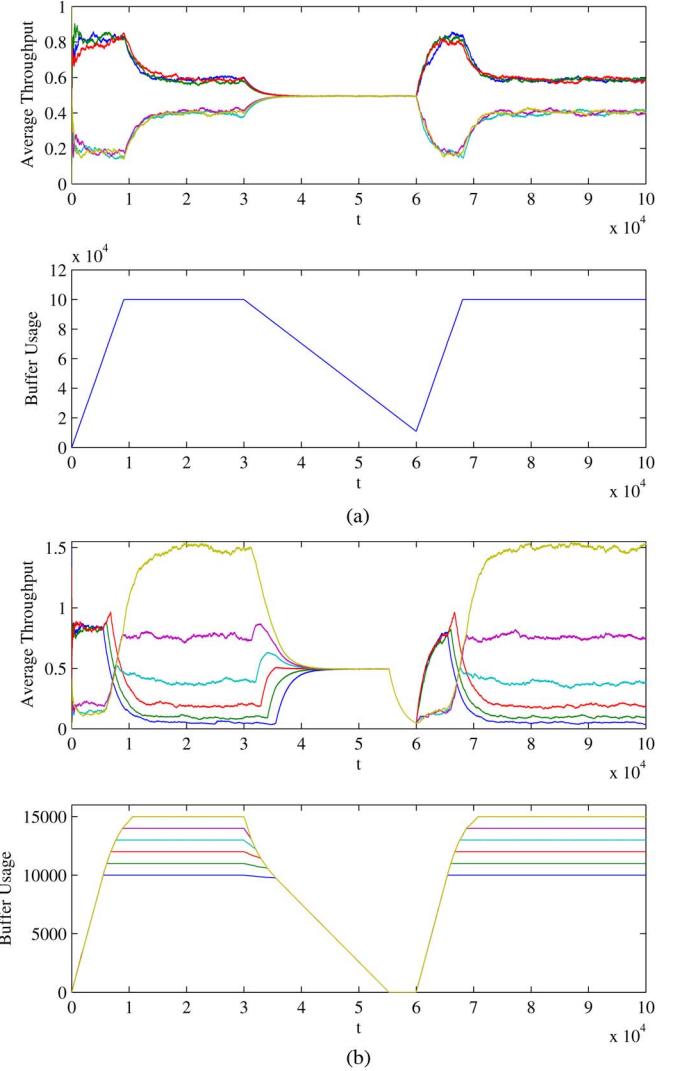


Fig. 6. The system behavior of a finite buffer network with GMW scheduler and Drop-Tail scheme. $\alpha = 1$, $b = 1$. (a) Shared Buffer, $B^{\max} = 10^5$. (b) Dedicated Buffer, $B_i^{\max} = 10\,000 + (i - 1) \times 1000$, $i = \{1, 2, 3, 4, 5, 6\}$.

The channel of each user is independent of each other, and has two states (G and B). The transmission rate of user i in state G and B are R_i^G and R_i^B , respectively. Assume that the probability of user i in state G is π_i^G , then $\pi_i^B = 1 - \pi_i^G$. The ergodic capacity region can be obtained as

$$\mathcal{C} = \left\{ \mathbf{r} : r_i \leq \sum_i t_i^G \pi_i^G R_i^G + t_i^B \pi_i^B R_i^B, \sum_i t_i^B + t_i^G \leq 1 \right\}.$$

The behavior of a temporarily overloaded system is illustrated in Fig. 6. Here we simulate a 6-user system in 100 000 time slots, and choose $\forall i, R_i^B = 1, R_i^G = 3, \pi_i^G = 1/2$. During the first 30 000 time slots and the last 40 000 time slots, the system has a Poisson arrival traffic with arrival rate λ , where $\lambda_1 = \lambda_2 = \lambda_3 = 8/3$ and $\lambda_4 = \lambda_5 = \lambda_6 = 2$. From time slot 30 001 to time slot 60 000, there is no traffic arrived in the system.

For the shared buffer case, by investigating Fig. 6(a), first the average throughput converges to a point determined by the infinite buffer case. After the time slot 10 000, as the buffer is

full, the average throughput converges to the point determined by the finite buffer case. After the time slot 30 000, since there is no new arrival traffic, the system can be stabilized and the average throughput converges to the same value, which is due to the symmetric channel. After the time slot 60 000, a similar pattern as that during time slots 1–30 000 can be found.

The results of the dedicated buffer case are shown in Fig. 6(b). The curves from top to down in the buffer usage figure are for user 1 to user 6, respectively. The average throughput exhibits a similar trend as that in the shared buffer case. The buffer usages of different users in the transient period (none of the buffer is full) are identical, which is a result of the symmetric channel assumption.

Overall, the simulation results have validated our analytical model and confirmed our analytical results and conclusions.

VIII. CONCLUSION AND FURTHER DISCUSSION

In this paper, we have studied the limiting properties of an overloaded multiuser wireless system with throughput-optimal scheduling. By studying a general throughput-optimal scheduling, we have found that certain results obtained in a system with special throughput-optimal scheduling is possibly universal. More specifically, we have shown that if the system is subject to infinite buffer assumption, all the queues in the network are unstable, but the average throughput and the scheduling function of queue length converges, respectively. By studying GMW and Log-Rule algorithms, we have found that the fairness of the corresponding system generally cannot be guaranteed. The Log-Rule can be viewed as a special GMW, and GMW can provide user differentiation by choosing parameters properly. If the buffer size is finite, then the buffer and queue management schemes play an important role on the network performance, and a proper design may alleviate the potential starvation problem.

There are several open issues left behind. First, we have assumed the achievable rate region $\mathcal{C}(t)$ is always a N -dimensional region. But in practice, if the channel is deep faded, the corresponding user may have zero achievable rate, and we cannot always increase the allocated rate of one user by decreasing the rates of other users. Second, the throughput-optimal scheduling considered is only queue-length based, which excludes the delay-driven throughput-optimal scheduling. Given the linear relationship between delay and queue length [14], the analysis for delay-driven throughput-optimal may have a similar result. Third, some queue-length based throughput-optimal scheduling algorithms, such as EXP-rule [5], are excluded from the discussion. How to extend our work to consider a more general throughput-optimal scheduling, such as the one proposed in [16] remains an open issue.

APPENDIX A PROOF OF THEOREM 1

Suppose that queue 1 is stable, and queue 2 is unstable. Then, we can find a B_1 such that for all t , $\Pr\{f_1(q_1(t)) < B_1\} > 1 - \epsilon_1$, where $\epsilon_1 > 0$. Because queue 2 is unstable, then for any $B > 0$, we have $\limsup_{t \rightarrow \infty} \Pr\{q_2(t) < B\} < \epsilon_2$,

where $\epsilon_2 > 0$. So we have $\liminf_{t \rightarrow \infty} \Pr\{q_2(t) > B\} > 1 - \epsilon_2$. Because f_2 is a strictly increasing continuous function, we have $\liminf_{t \rightarrow \infty} \Pr\{f_2(q_2(t)) > f_2(B)\} > 1 - \epsilon_2$. By choosing $B_2 = f_2(B)$, we have $\liminf_{t \rightarrow \infty} \Pr\{f_2(q_2(t)) > B_2\} > 1 - \epsilon_2$.

Suppose that the optimal solution for the following problem is $\eta^*(t)$,

$$\max_{\eta \in \mathcal{C}(t)} \sum_i w_i \eta_i,$$

where for all the i , $w_i > 0$. Because $\mathcal{C}(t)$ is a N -dimensional region, i.e., we can always increase the rate of one user by decreasing the rates of other users. Therefore, by increasing w_1 and decreasing w_2 , $\eta_1^*(t)$ will increase and $\eta_2^*(t)$ will decrease.

So we have, when $t \rightarrow \infty$, with probability $(1 - \epsilon_1)(1 - \epsilon_2)$, the rate allocated to user 1 is upper-bounded by $r_1^*(t)$, and $\mathbf{r}^*(t)$ is the solution to the following problem,

$$\max_{\eta \in \mathcal{C}(t)} B_1 \eta_1 + B_2 \eta_2 + f_3(q_3(t)) \eta_3 + \dots + f_N(q_N(t)) \eta_N. \quad (13)$$

Since for any B_2 and ϵ_2 , we have $\lim_{t \rightarrow \infty} \Pr\{f_2(q_2(t)) > B_2\} > 1 - \epsilon_2$. Consequently, for any given ϵ_1 and the corresponding B_1 , with any given ϵ_2 , we can choose a B_2 such that $B_2 \gg B_1$. Then based on (13), we have $r_1^*(t) \rightarrow 0$ as $B_2 \rightarrow \infty$ and $B_2 \gg B_1$.

We conclude that with probability $(1 - \epsilon_1)(1 - \epsilon_2)$, when $t \rightarrow \infty$, the average rate allocated to user 1 is upper-bounded by $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{\tau=t}^{t+T-1} r_1^*(\tau)$, where $\lim_{t \rightarrow \infty} r_1^*(t) \rightarrow 0$.

Since the average rate allocated to user 1 is always upper-bounded by the average achievable rate which is a finite value, with probability $1 - (1 - \epsilon_1)(1 - \epsilon_2)$, the average rate allocated to user 1 is upper-bounded by a finite value.

In summary, we can conclude that the average rate allocated to user 1 is upper-bounded by a value which approaches to 0 as the increment of time. As a result, queue 1 is not possible to be stable, which contradicts the assumption. Thus we proved Theorem 1.

APPENDIX B PROOF OF THEOREM 2

First we prove that the average throughput of the system converges.

For any $n > 0$ we can always find $C_{1,i}$ and $C_{2,i} > 0$ such that

$$q_i(t) - C_{2,i} \leq q_i(t+n) \leq q_i(t) + C_{1,i}.$$

Since f_i is a non-negative increasing continuous function, we have

$$f_i(q_i(t) - C_{2,i}) \leq f_i(q_i(t+n)) \leq f_i(q_i(t) + C_{1,i}).$$

Since q_i is unstable, with probability $1 - \epsilon$, for any $M > 0$, there exists a T such that for all $t > T$, we have $q_i(t) > M$. Further based on condition 2) of the scheduling algorithm, for any i and $0 < \sigma_i < 1$, for all $t > T$, we have

$$(1 - \sigma_i) f_i(q_i(t)) \leq f_i(q_i(t+n)) \leq (1 + \sigma_i) f_i(q_i(t)).$$

Then we have

$$\frac{(1 - \sigma_i)f_i(q_i(t))}{\sum_i(1 + \sigma_i)f_i(q_i(t))} \leq \bar{f}_i(\mathbf{q}(t + n)) \leq \frac{(1 + \sigma_i)f_i(q_i(t))}{\sum_i(1 - \sigma_i)f_i(q_i(t))}.$$

Define

$$\Delta_i(t, n) \triangleq \bar{f}_i(\mathbf{q}(t + n)) - \bar{f}_i(\mathbf{q}(t)),$$

and choose $\sigma_i = \sigma$, so we have

$$\left(\frac{1 - \sigma}{1 + \sigma} - 1\right) \bar{f}_i(\mathbf{q}(t)) \leq \Delta_i(t, n) \leq \left(\frac{1 + \sigma}{1 - \sigma} - 1\right) \bar{f}_i(\mathbf{q}(t)).$$

Equivalently, we have

$$|\Delta_i(t, n)| \leq \max\left(\frac{2\sigma}{1 + \sigma}, \frac{2\sigma}{1 - \sigma}\right) \bar{f}_i(\mathbf{q}(t)) = \frac{2\sigma}{1 - \sigma} \bar{f}_i(\mathbf{q}(t)).$$

Define

$$\delta_i \triangleq \frac{2\sigma}{1 - \sigma},$$

since $\bar{f}_i(\mathbf{q}(t)) \leq 1$, we have

$$|\Delta_i(t, n)| \leq \delta_i.$$

So for any δ_i and n we can find a corresponding σ that satisfies $|\Delta_i(t, n)| < \delta_i$.

In summary, with probability $1 - \epsilon$, for any i and δ_i , we can find a T such that for all $t > T$ and any $n > 0$,

$$|\bar{f}_i(\mathbf{q}(t + n)) - \bar{f}_i(\mathbf{q}(t))| \leq \delta_i,$$

and then we can conclude that for any i , $\bar{f}_i(\mathbf{q}(t))$ is a Cauchy sequence indexed by t , thus Cauchy converges in probability.

Suppose that $\bar{f}(\mathbf{q}(t))$ converges to \mathbf{w} . According to (2), $\mathbf{r}(t)$ converges to a solution to the following problem

$$\max_{\eta \in \mathcal{C}(t)} \sum_i w_i \eta_i,$$

which is only related to the capacity region in time slot t ($\mathcal{C}(t)$) and a weight vector (\mathbf{w}). Consequently, the convergence of the average throughput,

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \mathbf{r}(t_0 + t),$$

only requires the existence of the average capacity region, which is guaranteed by the assumption that the fading channel process is ergodic. Thus, the average throughput converges in probability, and we use μ to denote it.

Then we prove that μ is the solution to an optimization problem.

Since all the queues are unstable, with probability $1 - \epsilon$, for any $M > 0$, there exists a T such that for all $t > T$, we have $\min_i q_i(t) > M$. So for all $t > T$,

$$\mathbf{q}(t) = \mathbf{q}(t - 1) - \mathbf{r}(t - 1) + \mathbf{a}(t - 1),$$

by taking summation from T to t , we have

$$\begin{aligned} \mathbf{q}(t) &= \mathbf{q}(T) + a(T) + a(T + 1) + \dots + a(t - 1) \\ &\quad - (r(T) + r(T + 1) + \dots + r(t - 1)). \end{aligned}$$

Note that

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t - T} (a(T) + a(T + 1) + \dots + a(t - 1)),$$

$$\mu = \lim_{t \rightarrow \infty} \frac{1}{t - T} (r(T) + r(T + 1) + \dots + r(t - 1)),$$

so we have

$$\mathbf{q}(t) = (\lambda - \mu)t + \mu'(t) + \mathbf{q}(T) - (\lambda - \mu)T,$$

where

$$\lim_{t \rightarrow \infty} \frac{\mu'(t)}{t} = 0.$$

As $\mathbf{q}(T) - (\lambda - \mu)T$ is finite, hence, with probability $1 - \epsilon$, when $t > T$,

$$\begin{aligned} (1 - \sigma)f_i((\lambda_i - \mu_i)t + \mu'_i(t)) &\leq f_i(q_i(t)) \\ &\leq (1 + \sigma)f_i((\lambda_i - \mu_i)t + \mu'_i(t)), \end{aligned}$$

then based on the identical approach as to prove the convergence of $\bar{f}_i(\mathbf{q}(t))$, we can first prove the convergence of $\bar{f}_i((\lambda - \mu)t + \mu'(t))$ and then by using the squeeze theorem to prove that $\lim_{t \rightarrow \infty} \bar{f}_i(\mathbf{q}(t)) = \lim_{t \rightarrow \infty} \bar{f}_i((\lambda - \mu)t + \mu'(t))$ in probability. Therefore $\mathbf{r}(t)$ is the solution to the following problem,

$$\max_{\eta \in \mathcal{C}(t)} \sum_i \lim_{t \rightarrow \infty} \bar{f}_i((\lambda - \mu)t + \mu'(t)) \eta_i.$$

Then we can conclude that the average throughput μ converges in probability and is a solution to the following problem,

$$\max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i \lim_{t \rightarrow \infty} \bar{f}_i((\lambda - \mu)t + \mu'(t)) \bar{\eta}_i.$$

Next, we prove μ is unique.

The antiderivative of $f_i((\lambda_i - \mu_i)t + \mu'_i(t))$ is $-tF_i((\lambda_i - \mu_i)t + \mu'_i(t))$. Since $\lim_{t \rightarrow \infty} \sum_i f_i((\lambda_i - \mu_i)t + \mu'_i(t))\mu_i \geq \lim_{t \rightarrow \infty} \sum_i f_i((\lambda_i - \mu_i)t + \mu'_i(t))\bar{\eta}_i$, i.e.,

$$\lim_{t \rightarrow \infty} \sum_i f_i((\lambda_i - \mu_i)t + \mu'_i(t))(\mu_i - \bar{\eta}_i) \geq 0.$$

Furthermore because $f_i(x)$ is a strictly increasing continuous function, μ is the solution to the following problem [31]

$$\lim_{t \rightarrow \infty} \max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i F_i((\lambda_i - \bar{\eta}_i)t + \mu'_i(t)).$$

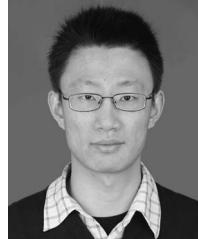
As $\lim_{t \rightarrow \infty} \mu'_i(t)/t = 0$, the above problem is further equivalent to

$$\lim_{t \rightarrow \infty} \max_{\bar{\eta} \in \mathcal{C}^\lambda} \sum_i F_i((\lambda_i - \bar{\eta}_i)t),$$

whose solution exists and is unique.

REFERENCES

- [1] X. Wang, G. B. Giannakis, and A. G. Marques, "A unified approach to QoS-guaranteed scheduling for channel-adaptive wireless networks," *Proc. IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec. 2007.
- [2] A. L. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation," *Oper. Res.*, vol. 53, no. 1, pp. 12–25, Jan. 2005.
- [3] X. Wang and L. Cai, "Stability region of opportunistic scheduling in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 4017–4027, Oct. 2014.
- [4] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.
- [5] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," in *American Mathematical Society Translations: Series 2*. Providence, RI, USA: Amer Mathematical Society, 2000.
- [6] A. L. Stolyar, "Large deviations of queues sharing a randomly time-varying server," *Queueing Syst.*, vol. 59, no. 1, pp. 1–35, May 2008.
- [7] V. J. Venkataraman and X. Lin, "On wireless scheduling algorithms for minimizing the queue-overflow probability," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 788–801, Jun. 2010.
- [8] B. Sadiq, S. J. Baek, and G. de Veciana, "Delay-optimal opportunistic scheduling and approximations: The log rule," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 405–418, Apr. 2011.
- [9] M. J. Neely, "Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, pp. 1188–1199, Oct. 2008.
- [10] K. Seong, R. Narasimhan, and J. M. Cioffi, "Queue proportional scheduling via geometric programming in fading broadcast channels," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1593–1602, Aug. 2006.
- [11] C. Zhou and G. Wunder, "Throughput-optimal scheduling with low average delay for cellular broadcast systems," *EURASIP J. Adv. Signal Process.*, vol. 2009, pp. 762050-1–762050-11, 2009.
- [12] K. Jagannathan, M. Markakis, E. Modiano, and J. N. Tsitsiklis, "Queue-length asymptotics for generalized max-weight scheduling in the presence of heavy-tailed traffic," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1096–1111, Apr. 2012.
- [13] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis, "Max-weight scheduling in queueing networks with heavy-tailed traffic," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 257–270, Feb. 2014.
- [14] M. Andrews *et al.*, "Scheduling in a queueing system with asynchronously varying service rates," *Probability Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, Apr. 2004.
- [15] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 411–424, Apr. 2005.
- [16] C. Zhou and G. Wunder, "A fundamental characterization of stability in broadcast queueing systems," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 1418–1422.
- [17] S. Meyn, "Stability and asymptotic optimality of generalized MaxWeight policies," *SIAM J. Control Optim.*, vol. 47, no. 6, pp. 3259–3294, Nov. 2008.
- [18] T. Bonald and J. Roberts, "Performance modeling of elastic traffic in overload," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, pp. 342–343, Jun. 2001.
- [19] S. Tekin, S. Andradóttir, and D. G. Down, "Dynamic server allocation for unstable queueing networks with flexible servers," *Queueing Syst.*, vol. 70, no. 1, pp. 45–79, Jan. 2012.
- [20] Y. Guo, E. Lefeber, Y. Nazarathy, G. Weiss, and H. Zhang, "Stability and performance for multi-class queueing networks with infinite virtual queues," *Queueing Syst. Theory Appl.*, vol. 76, no. 3, pp. 309–342, Mar. 2014.
- [21] L. Georgiadis and L. Tassiulas, "Optimal overload response in sensor networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2684–2696, Jun. 2006.
- [22] R. Egorova, S. Borst, and B. Zwart, "Bandwidth-sharing networks in overload," *Perform. Eval.*, vol. 64, no. 9–12, pp. 978–993, Oct. 2007.
- [23] D. Shah and D. Wischik, "Fluid models of congestion collapse in overloaded switched networks," *Queueing Syst.*, vol. 69, no. 2, pp. 121–143, Oct. 2011.
- [24] C. W. Chan, M. Armony, and N. Bambos, Fairness in Overloaded Parallel Queues, arXiv preprint arXiv:1011.1237v2, 2011.
- [25] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, "Throughput capacity of MANETs with power control and packet redundancy," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 3035–3047, Jun. 2013.
- [26] R. Jain, D.-M. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," Digital Equipment Corp., Hudson, MA, USA, DEC Res. Rep. TR-301, 1984.
- [27] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
- [28] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "On the stability of input-queued switches with speed-up," *IEEE/ACM Trans. Netw.*, vol. 9, no. 1, pp. 104–118, Feb. 2001.
- [29] X. Wang and L. Cai, "Proportional fair scheduling in hierarchical modulation aided wireless networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1584–1593, Apr. 2013.
- [30] X. Wang and L. Cai, "Resource allocation in a K-User wireless broadcast system with N-layer superposition coding," in *Proc. IEEE WCNC*, Shanghai, China, Apr. 2013, pp. 789–794.
- [31] D. P. Bertsekas, *Nonlinear Programming: 2nd Edition*. Belmont, MA, USA: Athena Scientific, 1999.
- [32] B. Radunovic and J.-Y. Le Boudec, "A Unified framework for max-min and min-max fairness with applications," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1073–1083, Oct. 2007.
- [33] L. Georgiadis, "Lexicographically optimal balanced networks," in *Proc. IEEE INFOCOM*, 2001, pp. 689–698.
- [34] C. Li, G. S. Paschos, L. Tassiulas, and E. Modiano, "Dynamic overload balancing in server farms," in *Proc. IFIP Netw.*, 2014, pp. 1–9.
- [35] S. Meyn, *Control Techniques Complex Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
- [36] V. G. Kulkarni, "Fluid models for single buffer systems," in *Frontiers Queueing*, J. H. Dshalalow, Ed. Boca Raton, FL, USA: CRC Press, 1998, pp. 321–338.
- [37] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [38] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis, "Max-weight scheduling in queueing networks with heavy-tailed traffic," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 257–270, Feb. 2014.



Xuan Wang (S'09) received the B.Eng. degree in information security and the M.S. degree in signal and information processing from Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2010, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada. His current research interests include scheduling, resource allocation, and cross-layer design in wireless networks.



Lin Cai (S'00–M'06–SM'10) received the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively. Since 2005, she has been an Assistant Professor and then an Associate Professor with the Department of Electrical and Computer Engineering, University of Victoria. Her research interests include wireless communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic over wireless, mobile, ad hoc, and sensor networks. She received the NSERC Discovery Accelerator Supplement Grant in 2010, the Best Paper Award of IEEE ICC 2008, and the Best Academic Paper Award of IEEE WCNC 2011. She has served as a Technical Program Committee Cochair for IEEE GLOBECOM'10 and GLOBECOM'13, and as the Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, EURASIP Journal on Wireless Communications and Networking, International Journal of Sensor Networks, and Journal of Communications and Networks.