Achieving Optimal Throughput and Near-Optimal Asymptotic Delay Performance in Multi-Channel Wireless Networks with Low Complexity: A Practical Greedy Scheduling Policy

Bo Ji, Gagan R. Gupta, Manu Sharma, Xiaojun Lin, and Ness B. Shroff

***Abstract*—In this paper, we focus on the scheduling problem in multi-channel wireless networks, e.g., the downlink of a single cell in fourth generation (4G) OFDM-based cellular networks. Our goal is to design practical scheduling policies that can achieve *provably good performance* in terms of both *throughput* and *delay*, at a *low complexity*. While a class of $O(n^{2.5}\log n)$-complexity hybrid scheduling policies are recently developed to guarantee both rate-function delay optimality (in the many-channel many-user *asymptotic* regime) and throughput optimality (in the general *non-asymptotic* setting), their practical complexity is typically high. To address this issue, we develop a simple greedy policy called *Delay-based Server-Side-Greedy (D-SSG)* with a *lower complexity* $2n^2+2n$, and rigorously prove that D-SSG not only achieves *throughput optimality*, but also guarantees *near-optimal asymptotic delay performance*. Specifically, we define the delay-violation probability as the steady-state probability that the largest packet waiting time in the system exceeds a certain fixed integer threshold $b>0$, and we study the rate-function (or decay-rate) of such delay-violation probability when the number of channels or users, n, goes to infinity. We show that the rate-function attained by D-SSG for any such threshold b, is no smaller than the maximum achievable rate-function by any scheduling policy for threshold $b-1$. Thus, we are able to achieve a reduction in complexity (from $O(n^{2.5}\log n)$ of the hybrid policies to $2n^2+2n$) with a minimal drop in the delay performance. More importantly, in practice, D-SSG generally has a substantially lower complexity than the hybrid policies that typically have a large constant factor hidden in the $O(\cdot)$ notation. Finally, we conduct numerical simulations to validate our theoretical results in various scenarios. The simulation results show that in all scenarios we consider, D-SSG not only guarantees a near-optimal rate-function, but also empirically has a similar delay performance to the rate-function delay-optimal policies.**

I. Introduction

In this paper, we consider the scheduling problem in a multi-channel wireless network, where the system has a large bandwidth that can be divided into multiple orthogonal sub-bands (or channels). A practically important example of such a multi-channel network is the downlink of a single cell of a fourth generation (4G) OFDM-based wireless cellular system (e.g., LTE and WiMax). In such a multi-channel system, a key challenge is *how to design efficient scheduling policies that can simultaneously achieve high throughput and low delay.* This problem becomes extremely critical in OFDM systems that are expected to meet the dramatically increasing demands from multimedia applications with more stringent Quality-of-Service (QoS) requirements (e.g., voice and video applications), and thus look for new ways to achieve higher data rates, lower latencies, and a much better user experience. Yet, an even bigger challenge is *how to design such high-performance scheduling policies at a low complexity.* For example, in the OFDM-based LTE systems, the *Transmission Time Interval (TTI)*, within which the scheduling decisions need to be made, is only one millisecond. On the other hand, there are hundreds of orthogonal channels that need to be allocated to hundreds of users. Hence, the scheduling decision has to be made within a very short scheduling cycle.

We consider a single-cell multi-channel system consisting of n channels and a proportionally large number of users, with intermittent connectivity between each user and each channel. We assume that the Base Station (BS) maintains a separate First-in First-out (FIFO) queue associated with each user, which buffers the packets for the user to download. A series of works studied the delay performance of scheduling policies in the large-queue asymptotic regime, where the buffer overflow threshold tends to infinity (see [1]–[4] and references therein). One potential difficulty of the large-queue asymptotic is that the estimates become accurate only when the queue-length or the delay becomes large. However, for a practical system that aims to serve a large number of users with more stringent delay requirements (as anticipated in the 4G systems), it is more important to ensure small queue-length and small delay [5]. Note that even in the wireline networks, there was a similar distinction between the large-buffer asymptotic and the many-source asymptotic [6], [7]. It was shown that the many-source asymptotic provides sharper estimates of the buffer violation probability when the queue-length threshold is not very large. Hence, the delay metric that we focus on in this paper is the *decay-rate* (or called the *rate-function* in large-deviations theory) of the steady-state probability that the largest packet waiting time in the system exceeds a certain fixed threshold when the number of users and the number of channels both go to infinity. (See Eq. (2) for the formal definition of rate-function.) We refer to this setting as the *many-channel many-user asymptotic regime*.

A number of recent works have considered a multi-channel system similar to ours, but looked at delay from different perspectives. A line of works focused on queue-length-based metrics: average queue length [8] or queue-length rate-function in the many-channel many-user asymptotic regime [5], [9]–[11]. In [8], the authors focused on minimizing cost functions

B. Ji is with AT&T Labs. M. Sharma is with Qualcomm Technologies Inc, and made contribution on this work while he was at Purdue University. X. Lin is with School of ECE at Purdue University. N. B. Shroff is with Departments of ECE and CSE at the Ohio State University. Emails: ji.33@osu.edu, gagan.gupta@iitdalumni.com, sharma50@purdue.edu, linx@ecn.purdue.edu, shroff.11@osu.edu.

This work was supported in part by ARO MURI grant W911NF-08-1-0238 and NSF grants CNS-1065136 and CNS-1012700 and CNS-0643145.

A preliminary version of this work was presented at the IEEE INFOCOM 2013, Turin, Italy, April, 2013.

over a finite horizon, which includes minimizing the expected total queue length as a special case. The authors showed that their goal can be achieved in two special scenarios: 1) a simple two-user system, and 2) systems where fractional server allocation is allowed. In [5], [9]–[11], delay performance is evaluated by the queue violation probability, and its associated rate-function, i.e., the asymptotic decay-rate of the probability that the largest queue length in the system exceeds a fixed threshold in the many-channel many-user asymptotic regime. Although [5] and [11] proposed scheduling policies that can guarantee both throughput optimality and rate-function optimality, there are still a number of important dimensions that have space for improvement. First, although the decay-rate of the queue violation probability may be mapped to that of the delay-violation probability when the arrival process is deterministic with a constant rate [4], this is not true in general, especially when the arrivals are correlated over time. Further, [12]–[14] have shown through simulations that good queue-length performance does not necessarily imply good delay performance. Second, their results on rate-function optimality strongly rely on the assumptions that the arrival process is *i.i.d. not only across users, but also in time*, and that per-user arrival at any time is no greater than the largest channel rate. Third, even under this more restricted model, the lowest complexity of their proposed rate-function-optimal algorithms is $O(n^3)$. For more general models, no algorithm with provable rate-function optimality is provided.

Similar to this paper, another line of work [12], [13], [15] proposed delay-based scheduling policies[1] and directly focused on the delay performance rather than the queue-length performance. The performance of delay is often harder to characterize, because the delay in a queueing system typically does not admit a Markovian representation. The problem becomes even harder in a multi-user system with fading channels and interference constraints, where the service rate for individual queues becomes more unpredictable. In [12], [13], the authors developed a scheduling policy called Delay Weighted Matching (DWM), which maximizes the sum of the delay of the scheduled packets in each time-slot. It has been shown in [12], [13], [15] that DWM is not only throughput-optimal, but also rate-function delay-optimal (i.e., maximizing the *delay rate-function*, rather than the *queue-length rate-function* as considered in [5], [9]–[11]). Moreover, the authors of [13] used the derived rate-function of DWM to develop a simple threshold policy for admission control when the number of users scales linearly with the number of channels in the system. However, DWM incurs a high complexity $O(n^5)$, which renders it impractical for modern OFDM systems with many channels and users (e.g., on the order of hundreds). In [15], the authors proposed a class of hybrid scheduling policies with a much lower complexity $O(n^{2.5}\log n)$, while still guaranteeing both throughput optimality and rate-function delay optimality (with an additional minor technical assumption). However, the practical complexity of the hybrid policies is still high as the constant factor hidden in the $O(\cdot)$ notation is typically large due to the required two-stage scheduling operations and the operation of computing a maximum-weight matching in the first stage. Hence, scheduling policies with an even lower (both theoretical and practical) complexity are needed in the multi-user multi-channel systems.

This leads to the following natural but important questions: *Can we find scheduling policies that have a significantly lower complexity, with comparable or only slightly worse performance? How much complexity can we reduce, and how much performance do we need to sacrifice?* In this paper, we answer these questions positively. Specifically, we develop a *low-complexity* greedy policy that achieves both *throughput optimality* and *rate-function near-optimality*.

We summarize our main contributions as follows.

First, we propose a greedy scheduling policy, called *Delay-based Server-Side-Greedy (D-SSG)*. D-SSG, in an iterative manner, allocates servers one-by-one to serve a connected queue that has the largest head-of-line (HOL) delay. We rigorously prove that D-SSG not only achieves throughput optimality, but also guarantees a near-optimal rate-function. Specifically, the rate-function attained by D-SSG for any fixed integer threshold $b > 0$, is *no smaller than the maximum achievable rate-function by any scheduling policy for threshold* $b-1$. We obtain this result by comparing D-SSG with a new *Greedy Frame-Based Scheduling (G-FBS)* policy that can exploit a key property of D-SSG. We show that G-FBS policy guarantees a near-optimal rate-function, and that D-SSG dominates G-FBS in every sample-path. *To the best of our knowledge, this is the first work that shows a near-optimal rate-function in the above form, and hence we believe that our proof technique is of independent interest.* Also, we remark that the gap between the near-optimal rate-function attained by D-SSG and the optimal rate-function is likely to be quite small. (See Section IV-C for detailed discussion.)

D-SSG is a very simple policy and has a *low complexity* $2n^2+2n$. Note that the queue-length-based counterpart of D-SSG, called Q-SSG, has been studied in [9], [10]. However, there the authors were only able to prove a positive (queue-length) rate-function for restricted arrival processes that are *i.i.d.* not only across users, but also in time. In contrast, we show that D-SSG achieves a rate-function that is not only positive but also near-optimal, for more general arrival processes. Thus, we are able to achieve a reduction in complexity (from $O(n^{2.5}\log n)$ of the hybrid policies [15] to $2n^2+2n$) with a minimal drop in the delay performance. More importantly, the practical complexity of D-SSG is substantially lower than that of the hybrid policies since we can precisely bound the constant factor in its complexity.

Further, we conduct simulations to validate our analytical results in various scenarios. The simulation results show that in all scenarios we consider, D-SSG not only guarantees a near-optimal rate-function, but also empirically has a similar delay performance to the rate-function delay-optimal policies.

The remainder of the paper is organized as follows. In Section II, we describe the details of our system model and performance metrics. In Section III, we derive an upper bound on the rate-function that can be achieved by any scheduling

[1]Delay-based policies were first introduced in [16] for scheduling problems in Input-Queued switches, and were later studied for wireless networks [14], [17]–[22]. Please see [14] and references therein for more discussions on the history and the recent development of delay-based scheduling policies.

policy. Then, in Section IV, we present our main results on throughput optimality and near-optimal rate-function for our proposed low-complexity greedy policy. Further, we conduct numerical simulations in Section V. Finally, we make concluding remarks in Section VI.

II. System Model

We consider a discrete-time model for the downlink of a single-cell multi-channel wireless network with n orthogonal channels and n users. In each time-slot, a channel can be allocated only to one user, but a user can be allocated with multiple channels simultaneously. As in [5], [9]–[13], [15], for ease of presentation, we assume that the number of users is equal to the number of channels. (If the number of users scales linearly with the number of channels, the rate-function delay analysis follows similarly. However, an admission control policy needs to be carefully designed if the number of users becomes too large [13].) We let Q_i denote the FIFO queue associated with the i-th user, and let S_j denote the j-th server[2]. We consider the *i.i.d.* ON-OFF channel model under which the connectivity between each queue and each server changes between ON and OFF from time to time. We also assume unit channel capacity, i.e., at most one packet from Q_i can be served by S_j when Q_i and S_j are connected. Let $C_{i,j}(t)$ denote the connectivity between queue Q_i and server S_j in time-slot t. Then, $C_{i,j}(t)$ can be modeled as a Bernoulli random variable with a parameter $q \in (0,1)$, i.e.,

$$C_{i,j}(t) = \begin{cases} 1, & \text{with probability } q, \\ 0, & \text{with probability } 1-q. \end{cases}$$

We assume that all the random variables $C_{i,j}(t)$ are *i.i.d.* across all the variables i, j and t. Such a network can be modeled as a multi-queue multi-server system with stochastic connectivity, as shown in Fig. 1. Further, we assume that the perfect channel state information (i.e., whether each channel is ON or OFF for each user in each time-slot) is known at the BS. This is a reasonable assumption in the downlink scenario of a single cell in a multi-channel cellular system with dedicated feedback channels.

As in the previous works [5], [8], [9], [12], [13], [15], the above *i.i.d.* ON-OFF channel model is a simplification, and is assumed only for the analytical results. The ON-OFF model is a good approximation when the BS transmits at a fixed achievable rate if the SINR level is above a certain threshold at the receiver, and does not transmit successfully otherwise. The sub-bands being *i.i.d.* is a reasonable assumption when the channel width is larger than the coherence bandwidth of the environment. Moreover, we believe that our results obtained for this channel model can provide useful insights for more general models. Indeed, we will show through simulations that our proposed greedy policies also perform well in more general models, e.g., accounting for heterogeneous (near- and far-)users and time-correlated channels. Further, we will briefly discuss how to design efficient scheduling policies in general scenarios towards the end of this paper.

[2]Throughout this paper, we use the terms "user" and "queue" interchangeably, and use the terms "channel" and "server" interchangeably.

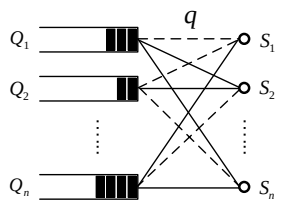


Fig. 1. System model. The connectivity between each pair of queue Q_i and server S_j is "ON" (denoted by a solid line) with probability q, and "OFF" (denoted by a dashed line) otherwise.

We present more notations used in this paper as follows. Let $A_i(t)$ denote the number of packet arrivals to queue Q_i in time-slot t. Let $A(t) = \sum_{i=1}^{n} A_i(t)$ denote the cumulative arrivals to the entire system in time-slot t, and let $A(t_1, t_2) = \sum_{\tau=t_1}^{t_2} A(\tau)$ denote the cumulative arrivals to the system from time t_1 to t_2. We let λ_i denote the mean arrival rate to queue Q_i, and let $\lambda \triangleq [\lambda_1, \lambda_2, \ldots, \lambda_n]$ denote the arrival rate vector. We assume that packets arrive at the beginning of a time-slot, and depart at the end of a time-slot. We use $Q_i(t)$ to denote the length of queue Q_i at the beginning of time-slot t immediately after packet arrivals. Queues are assumed to have an infinite buffer capacity. Let $Z_{i,l}(t)$ denote the delay (or waiting time) of the l-th packet at queue Q_i at the beginning of time-slot t, *which is measured from the time when the packet arrived to queue Q_i until the beginning of time-slot t*. Note that at the end of each time-slot, the packets that are still present in the system will have their delays increased by one due to the elapsed time. Further, let $W_i(t) = Z_{i,1}(t)$ (or $W_i(t) = 0$ if $Q_i(t) = 0$) denote the HOL delay of queue Q_i at the beginning of time-slot t. Finally, we define $(x)^+ \triangleq \max(x, 0)$, and use $\mathbb{1}_{\{\cdot\}}$ to denote the indicator function.

We now state the assumptions on the arrival processes. The throughput analysis is carried out under Assumption 1 only, which is mild and has also been used in [15], [19].

Assumption 1: For each user $i \in \{1, 2, \ldots, n\}$, the arrival process $A_i(t)$ is an irreducible and positive recurrent Markov chain with countable state space, and satisfies the Strong Law of Large Numbers: That is, with probability one,

$$\lim_{t\to\infty} \frac{\sum_{\tau=0}^{t-1} A_i(\tau)}{t} = \lambda_i. \tag{1}$$

We also assume that the arrival processes are mutually independent across users (which can be relaxed for throughput analysis as discussed in [19]).

The rate-function delay analysis is carried out under the following two assumptions, which have also been used in the previous works [12], [13], [15].

Assumption 2: There exists a finite L such that $A_i(t) \le L$ for any i and t, i.e., instantaneous arrivals are bounded.

Assumption 3: The arrival processes are *i.i.d.* across users, and $\lambda_i = p$ for any user i. Given any $\epsilon > 0$ and $\delta > 0$, there exist $T > 0$, $N > 0$, and a positive function $I_B(\epsilon, \delta)$ independent of n and t such that

$$\mathbb{P}\Big(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{|\sum_{i=1}^{n} A_i(\tau) - pn| > \epsilon n\}}}{t} > \delta\Big) < \exp(-ntI_B(\epsilon, \delta)),$$

for all $t > T$ and $n > N$.

Assumption 2 requires that the arrivals in each time-slot have bounded support, which is indeed true for practical systems. Assumption 3 is also very general, and can be viewed as a result of the statistical multiplexing effect of a large number of sources. Assumption 3 holds for *i.i.d.* arrivals and arrivals driven by two-state Markov chains (that can be correlated over time) as two special cases (see Lemmas 2 and 3 of [13]).

A. Performance Objectives

In this paper, we consider two performance metrics: 1) the *throughput* and 2) the steady-state probability that the largest packet delay in the system exceeds a certain fixed threshold, and its associated *rate-function* in the many-channel many-user asymptotic regime.

We first define the *optimal throughput region* (or *stability region*) of the system for any fixed integer $n > 0$ under Assumption 1. As in [19], a stochastic queueing network is said to be *stable* if it can be described as a discrete-time countable Markov chain and the Markov chain is stable in the following sense: The set of positive recurrent states is nonempty, and it contains a finite subset such that with probability one, this subset is reached within finite time from any initial state. When all the states communicate, stability is equivalent to the Markov chain being positive recurrent [23]. The *throughput region* of a scheduling policy is defined as the set of arrival rate vectors for which the network remains stable under this policy. Then, the *optimal throughput region* is defined as the union of the throughput regions of all possible scheduling policies, which is denoted by Λ^*. A scheduling policy is *throughput-optimal*, if it can stabilize any arrival rate vector strictly inside Λ^*. For more discussions on the optimal throughput region Λ^* in our multi-channel systems, please refer to [15].

Next, we consider the steady-state probability that the largest packet delay in the system exceeds a certain fixed threshold, and its associated *rate-function* in the many-channel many-user asymptotic regime. Assuming that the system is stationary and ergodic, let $W(0) \triangleq \max_{1\le i\le n} W_i(0)$ denote the largest HOL delay over all the queues (i.e., the largest packet delay in the system) in the steady state, and then we define rate-function $I(b)$ as the decay-rate of the probability that $W(0)$ exceeds any fixed integer threshold $b \ge 0$, as the system size n goes to infinity, i.e.,

$$I(b) \triangleq \lim_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b). \tag{2}$$

Note that once we know this rate-function, we can then estimate the delay-violation probability using $\mathbb{P}(W(0) > b) \approx \exp(-nI(b))$. The estimate tends to be more accurate as n becomes larger. Clearly, for systems with a large n, a larger value of the rate-function implies a better delay performance, i.e., a smaller probability that the largest packet delay in the system exceeds a certain threshold. As in [12], [13], [15], we define the *optimal rate-function* as the maximum achievable rate-function over all possible scheduling policies, which is denoted by $I^*(b)$. A scheduling policy is *rate-function delay-optimal* if it achieves the optimal rate-function $I^*(b)$ for any fixed integer threshold $b \ge 0$.

III. An Upper Bound on The Rate-Function

In this section, we derive an upper bound of the rate-function for all scheduling policies.

Let $I_{AG}(t,x)$ denote the asymptotic decay-rate of the probability that in any interval of t time-slots, the total number of arrivals is greater than $n(t+x)$, as n tends to infinity, i.e.,

$$I_{AG}(t,x) \triangleq \liminf_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(A(-t+1,0) > n(t+x)).$$

Let $I_{AG}(x)$ be the infimum of $I_{AG}(t,x)$ over all $t > 0$, i.e.,

$$I_{AG}(x) \triangleq \inf_{t>0} I_{AG}(t,x).$$

Also, we define $I_X \triangleq \log \frac{1}{1-q}$.

Theorem 1: Given the system model described in Section II, for any scheduling algorithm, we have

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b)$$
$$\le \min\{(b+1)I_X, \min_{0\le c\le b}\{I_{AG}(b-c) + cI_X\}\} \triangleq I_U(b).$$

Theorem 1 can be shown by considering two types of events that lead to the delay-violation event $\{W(0) > b\}$ no matter how packets are scheduled, and computing their probabilities and decay-rates. In the above expression of $I_U(b)$, the first term $(b+1)I_X$ is due to sluggish services, which corresponds to the event that a queue with at least one packet is disconnected from all of the n servers for consecutive $b+1$ time-slots. The second term $\min_{0\le c\le b}\{I_{AG}(b-c) + cI_X\}$ is due to both bursty arrivals and sluggish services, where $I_{AG}(b-c)$ corresponds to the event that the arrivals are too bursty during the interval of $[-t-b, -b-1]$ such that at the beginning of time-slot $-c$ for $c \le b$, there exists at least one packet remaining in the system, say queue Q_1. Then, the term cI_X corresponds to the event that the services are too sluggish such that queue Q_1 is disconnected from all of the n servers for the following consecutive c time-slots. Clearly, both of the above events will lead to the delay-violation event $\{W(0) > b\}$ under all scheduling policies. We provide the detailed proof in Appendix A.

Remark: Theorem 1 implies that $I_U(b)$ is an upper bound on the rate-function that can be achieved by any scheduling policy. Hence, even for the optimal rate-function $I^*(b)$, we must have $I^*(b) \le I_U(b)$ for any fixed integer threshold $b \ge 0$.

Note that our derived upper bound $I_U(b)$ is strictly positive in the cases of interest. For example, when $L = 1$, it has been shown in [15] that the optimal rate-function is $I^*(b) = (b+1)\log\frac{1}{1-q}$, and thus $I_U(b) \ge I^*(b) > 0$ for all integer $b \ge 0$. This holds for general arrival processes under Assumptions 2 and 3, including two special cases of *i.i.d.* Bernoulli arrivals and two-state Markov chain driven arrivals. When $L > 1$, we can show that $I_U(b)$ is strictly positive for the special case of *i.i.d.* 0-L arrivals with feasible arrival rates (please refer to our online technical report [24]); further, in Section V, our simulation results (Fig. 2) also demonstrate that the rate-function attained by D-SSG is strictly positive under two-state Markov chain driven arrivals.

IV. Delay-based Server-Side-Greedy (D-SSG)

In [15], it has been shown that a class of two-stage hybrid policies can achieve both throughput optimality and rate-function delay optimality at a lower complexity $O(n^{2.5}\log n)$

(compared to $O(n^5)$ of DWM). The hybrid policies are constructed by combining certain throughput-optimal policies with a rate-function delay-optimal policy DWM-n (where n is the number of users or channels), which in each time-slot maximizes the sum of the delay of the scheduled packets among the n oldest packets in the system. For example, DWM-n combined with the Delay-based MaxWeight Scheduling (D-MWS) policy [15], [19], [20] yields a $O(n^{2.5}\log n)$ complexity hybrid policy, called the DWM-n-MWS policy.

The above result leads to the following important questions: *Is it possible to develop scheduling policies with an even lower complexity, while achieving comparable or only slightly worse performance? If so, how much complexity can we reduce, and how much performance do we need to sacrifice?* In this section, we answer these questions positively. We first develop a greedy scheduling policy called *Delay-based Server-Side-Greedy (D-SSG)* with an even lower complexity $2n^2+2n$. Under D-SSG, each server iteratively chooses to serve a connected queue that has the largest HOL delay. Then, we show that D-SSG not only achieves throughput optimality, but also guarantees a near-optimal rate-function. Hence, D-SSG achieves a reduction in complexity (from $O(n^{2.5}\log n)$ of the hybrid policies to $2n^2+2n$) with a minimal drop in the delay performance. More importantly, the practical complexity of D-SSG is substantially lower than that of the hybrid policies.

A. Algorithm Description

Before we describe the detailed operations of D-SSG, we would like to remark on the D-MWS policy in our multi-channel system, due to the similarity between D-MWS and D-SSG. Under D-MWS, each server chooses to serve a queue that has the largest HOL delay (among all the queues connected to this server). Note that D-MWS is not only throughput-optimal, but also has a low complexity $O(n^2)$. However, in [15] it has been shown that D-MWS suffers from poor delay performance. (Specifically, D-MWS yields a rate-function of zero in certain scenarios, e.g., with *i.i.d.* 0-1 arrivals). The reason is that under D-MWS, each server chooses to serve a connected queue that has the largest HOL delay without accounting for the decisions of the other servers. This way of allocating servers leads to an unbalanced schedule. That is, only a small fraction of the queues get served in each time-slot. This inefficiency leads to poor delay performance.

Now, we describe the operations of our proposed D-SSG policy. D-SSG is similar to D-MWS, in the sense that it also allocates each server to a connected queue that has the largest HOL delay. However, the key difference is that, instead of allocating the servers all at once as in D-MWS, D-SSG allocates the servers one-by-one, accounting for the scheduling decisions of the servers that are allocated earlier. We will show that *this critical difference results in a substantial improvement in the delay performance.*

We present some additional notations, and then specify the detailed operations of D-SSG. In each time-slot, there are n rounds, and in each round, one of the remaining servers is allocated. Let $Q_i^k(t)$, $Z_{i,l}^k(t)$ and $W_i^k(t)=Z_{i,1}^k(t)$ (or $W_i^k(t)=0$ if $Q_i^k(t)=0$) denote the length of queue Q_i, the delay of the l-th packet of Q_i, and the HOL delay of Q_i after $k\geq 1$ rounds of server allocation in time-slot t, respectively. In particular, we have $Q_i^0(t)=Q_i(t)$, $Z_{i,l}^0(t)=Z_{i,l}(t)$, and $W_i^0(t)=W_i(t)$. Let $\mathcal{S}_j(t)$ denote the set of queues being connected to server S_j in time-slot t, i.e., $\mathcal{S}_j(t)=\{1\leq i\leq n \mid C_{i,j}(t)=1\}$. Let $\Gamma_j^k(t)$ denote the set of indices of the queues that are connected to server S_j in time-slot t and that have the largest HOL delay at the beginning of the k-th round in time-slot t, i.e., $\Gamma_j^k(t)\triangleq\{i\in\mathcal{S}_j(t)\mid W_i^{k-1}(t)=\max_{l\in\mathcal{S}_j(t)}W_l^{k-1}(t)\}$. Let $i(j,t)$ denote the index of queue that is served by server S_j in time-slot t under D-SSG.

Delay-based Server-Side-Greedy (D-SSG) policy: In each time-slot t,

1) Initialize $k=1$.
2) In the k-th round, allocate server S_k to serve queue $Q_{i(k,t)}$, where $i(k,t)=\min\{i\mid i\in\Gamma_k^k(t)\}$. That is, in the k-th round, the k-th server S_k is allocated to serve the connected queue that has the largest HOL delay, breaking ties by picking the queue with the smallest index if there are multiple such queues. Then, update the length of $Q_{i(k,t)}$ to account for service, i.e., set $Q_{i(k,t)}^k(t)=\left(Q_{i(k,t)}^{k-1}(t)-C_{i(k,t),k}(t)\right)^+$ and $Q_i^k(t)=Q_i^{k-1}(t)$ for all $i\neq i(k,t)$. Also, update the HOL delay of $Q_{i(k,t)}$ to account for service, i.e., set $W_{i(k,t)}^k(t)=Z_{i(k,t),1}^k(t)=Z_{i(k,t),2}^{k-1}(t)$ if $Q_{i(k,t)}^k(t)>0$, and $W_{i(k,t)}^k(t)=0$ otherwise, and set $W_i^k(t)=W_i^{k-1}(t)$ for all $i\neq i(k,t)$.
3) Stop if k equals n. Otherwise, increase k by 1 and repeat step 2.

Remark: From the above operations, it can be observed that in each round, D-SSG aims to allocate the available server with the smallest index. Further, when there are multiple queues that are connected to the considered server and that have the largest HOL delay, D-SSG favors the queue with the smallest index. We specify such tie-breaking rules for ease of analysis only. In practice, we can break ties arbitrarily.

We highlight that D-SSG has a low complexity of $2n^2+2n$ due to the following operations. Assume that each packet contains the information of its arriving time. At the beginning of each time-slot, it requires n addition operations to update the HOL delay of each of the n queues (i.e., increasing it by one). In each round k, it takes n time to check the connectivity between server S_k and the n queues, another up to n time to find the connected queue with the largest HOL delay, and one more basic operation to update the HOL delay of the queue chosen by server S_k. Since there are n rounds, the overall complexity is $n+n(n+n+1)=2n^2+2n$.

Note that the queue-length-based counterpart of D-SSG, called Q-SSG, has been studied in [9], [10]. Under Q-SSG, each server iteratively chooses to serve a connected queue that has the largest length. It has been shown that Q-SSG not only achieves throughput optimality, but also guarantees a *positive (queue-length)* rate-function. However, their results have the following limitations: 1) a positive rate-function may not be good enough, since the gap between the guaranteed

rate-function and the optimal is unclear; 2) good queue-length performance does not necessarily translate into good delay performance; 3) their analysis was only carried out for restricted arrival processes that are *not only i.i.d. across users, but also in time*. In contrast, in this section we will show that D-SSG achieves a rate-function that is *not only positive but also near-optimal* (in the sense of (3)) for more general arrival processes, while guaranteeing throughput optimality.

B. Throughput Optimality

We first establish throughput optimality of D-SSG in general non-asymptotic settings with any fixed value of n. Note that in Section IV-C, we will analyze the delay performance of D-SSG in the asymptotic regime, where n goes to infinity. Hence, even if the convergence rate of the delay rate-function is fast (as is typically the case), the throughput performance may still be poor for small to moderate values of n. As a matter of fact, for a fixed n, a rate-function delay-optimal policy (e.g., DWM-n) may not even be throughput-optimal [15]. To this end, we first focus on studying the throughput performance of D-SSG in general non-asymptotic settings.

We remark that the throughput performance of scheduling policies have been extensively studied in various settings, including the multi-channel systems that we consider in this paper. Specifically, for such multi-channel systems, [15] proposed a class of Maximum Weight in the Fluid limit (MWF) policies and proved throughput-optimality of the MWF policies in very general settings (under Assumption 1). The key insight is that to achieve throughput-optimality in such multi-channel systems, it is sufficient for each server to choose a connected queue with a large enough weight (i.e., queue-length or delay) such that this queue has the largest weight in the fluid limit [25].

Next, we prove that D-SSG is throughput-optimal in general non-asymptotic settings (for a system with any fixed value of n) by showing that D-SSG is an MWF policy.

Theorem 2: D-SSG policy is throughput-optimal under Assumption 1.

The proof of Theorem 2 is straightforward. Hence, we omit the proof and provide it in our online technical report [24].

C. Near-optimal Asymptotic Delay Performance

In this subsection, we present our main result on the near-optimal rate-function. We first define near-optimal rate-function, and then evaluate the delay performance of D-SSG.

A policy $\mathbf{P}$ is said to achieve *near-optimal rate-function* if the delay rate-function $I(b)$ attained by policy $\mathbf{P}$ for any fixed integer threshold $b > 0$, is no smaller than $I^*(b-1)$, the optimal rate-function for threshold $b-1$. That is,

$$I(b) = \liminf_{n\to\infty} \frac{-1}{n} \log \mathbb{P}\left(W(0) > b\right) \geq I^*(b-1). \quad (3)$$

We next present our main result of this paper in the following theorem, which states that D-SSG achieves a near-optimal rate-function.

Theorem 3: Under Assumptions 2 and 3, D-SSG achieves a near-optimal rate-function, as given in Eq. (3).

We prove Theorem 3 by the following strategy: 1) motivated by a key property of D-SSG (Lemma 1), we propose the Greedy Frame-Based Scheduling (G-FBS) *policy, which is a variant of the FBS policy [12], [13] that has been shown to be rate-function delay-optimal in some cases; 2) show that G-FBS achieves a near-optimal rate-function (Theorem 4); 3) prove a dominance property of D-SSG over G-FBS. Specifically, in Lemma 2, we show that for any given sample path, by the end of each time-slot, D-SSG has served every packet that G-FBS has served.*

We now present a crucial property of D-SSG in Lemma 1, which is the key to proving a near-optimal rate-function for D-SSG.

Lemma 1: Consider a set of n packets satisfying that no more than $2H$ packets are from the same queue, where $H > 4$ is any integer constant independent of n. Consider any strictly increasing function $f(n)$ such that $f(n) < \frac{n}{2}$ and $f(n) \in o(n/\log^2 n)$. Suppose that D-SSG is applied to schedule these n packets. Then, there exists a finite integer $N_X > 0$ such that for all $n \geq N_X$, with probability no smaller than $1 - 2(1-q)^{n - f(n)\log^2 n}$, D-SSG schedules at least $n - H\sqrt{n}$ packets, including the oldest $f(n)$ packets among the n packets.

To prove Lemma 1 and thus near-optimal rate-function of D-SSG (Theorem 3), we introduce another greedy scheduling policy called *Delay-based Queue-Side-Greedy (D-QSG)* and a sample-path equivalence property between D-QSG and D-SSG (Lemma 3). Please refer to Appendix B for details.

We provide the proof of Lemma 1 in Appendix C, and explain the importance of Lemma 1 as follows. We first recall how DWM is shown to be rate-function delay-optimal (for some cases) in [12], [13]. Specifically, the authors of [12], [13] compare DWM with another policy FBS. In FBS, packets are filled into frames with size $n - H$ in a First-Come First-Serve (FCFS) manner such that no two packets in the same frame have a delay difference larger than h time-slots, where $h > 0$ is a suitably chosen constant independent of n and $H = Lh$. The FBS policy attempts to serve the entire HOL frame whenever possible. The authors of [12], [13] first establish the rate-function optimality of the FBS policy. Then, by showing that DWM dominates FBS (i.e., DWM will serve the same packets in the entire HOL frame whenever possible), the delay optimality of DWM then follows.

However, this comparison approach will not work directly for D-SSG. In order to serve all packets in a frame whenever possible, one would need certain back-tracking (or rematching) operations as in a typical maximum-weight matching algorithm like DWM. For a simple greedy algorithm like D-SSG that does not do back-tracking, it is unlikely to attain the same probability of serving the entire frame. In fact, even if we reduce the maximum frame size to $n - H\sqrt{n}$, we are still unable to show that D-SSG can serve the entire frame with a sufficiently high probability. Thus, we cannot compare D-SSG with FBS as in [12], [13].

Fortunately, Lemma 1 provides an alternate avenue. Specifically, for a set of n packets, even though D-SSG may not serve any *given* subset of $n - H\sqrt{n}$ packets with a sufficiently high probability, it will serve *some* subset of $n - H\sqrt{n}$ packets with a sufficiently high probability. Further, this subset must contain

the oldest $H\sqrt{n}$ packets for a large n, if we choose $f(n)$ in Lemma 1 such that $f(n) > H\sqrt{n}$ for large n. Note that D-SSG still leaves (at most) $H\sqrt{n}$ packets to the next time-slot. If we can ensure that in the next time-slot, D-SSG serves all of these $H\sqrt{n}$ leftover packets, we would then at worst suffer an additional one-time-slot delay. Indeed, Lemma 1 guarantees this with high probability. Intuitively, we would then be able to show that D-SSG attains a near-optimal delay rate-function as given in Eq. (3).

To make this argument rigorous, we next compare D-SSG with a new policy called **Greedy Frame-Based Scheduling (G-FBS)**. Note that G-FBS is only for assisting our analysis, and will not be used as an actual scheduling algorithm. We first fix a properly chosen parameter $h > 0$. In the G-FBS policy, packets are grouped into frames satisfying the following requirements: 1) No two packets in the same frame have a delay difference larger than h time-slots. This guarantees that in a frame, no more than $H = Lh$ packets from the same queue can be filled into a single frame; 2) Each frame has a capacity of $n_0 = n - H\sqrt{n}$ packets, i.e., at most n_0 packets can be filled into a frame; 3) As packets arrive to the system in each time-slot, the frames are created by filling the packets sequentially. Specifically, packets that arrive earlier are filled into the frame with a higher priority, and packets from queues with a smaller index are filled with a higher priority when multiple packets arrive in the same time-slot. Once any of the above requirements is violated, the current frame will be closed and a new frame will be open. We also assume that there is a "leftover" frame, called *L-frame* for simplicity, with a capacity of $H\sqrt{n}$ packets. The L-frame is for storing the packets that were not served in the previous time-slot and were carried over to the current time-slot. At the beginning of each time-slot, we combine the HOL frame and the L-frame into a "super" frame, called *S-frame* for simplicity, with a capacity of n packets. It is easy to see that in the S-frame, no more than $2H$ packets are from the same queue. Note that if there are less than n packets in the S-frame, we can artificially add some dummy packets with a delay of zero at the end of the S-frame so that the S-frame is fully filled, but still need to guarantee that no more than $2H$ packets from the same queue can be filled into the S-frame. In each time-slot, G-FBS runs the D-SSG policy, but restricted to only the n packets of the S-frame. We call it a *success*, if D-SSG can schedule at least n_0 packets, including the oldest $f(n)$ packets, from the S-frame, where $f(n) < \frac{n}{2}$ is any function that satisfies that $f(n) \in o(n/\log^2 n)$ and $f(n) \in \omega(\sqrt{n})$. In each time-slot, if a success does not occur, then no packets will be served. When there is a success, the G-FBS policy serves all the packets that are scheduled by D-SSG restricted to the S-frame in that time-slot. *Lemma 1 implies that in each time-slot, a success occurs with probability at least* $1 - 2(1-q)^{n - f(n)\log^2 n}$. A success serves all packets from the S-frame, except for at most $H\sqrt{n} = n - n_0$ packets, and these served packets include the oldest $f(n)$ packets. The packets that are not served will be stored in the L-frame, and carried over to the next time-slot (except for the dummy packets, which will be discarded).

Remark: Although G-FBS is similar to FBS policy [12], [13], it exhibits a key difference from FBS. In the FBS policy, in each time-slot, either an entire frame (i.e., all the packets in the frame) will be completely served or none of its packets will be served. Hence, it does not allow packets to be carried over to the next time-slot. In contrast, G-FBS allows leftover packets and is thus more flexible in serving frames. This property is the key reason that we can use lower-complexity policies like D-SSG. On the other hand, it leads to a small gap between the rate-functions achieved by G-FBS and delay-optimal policies (e.g., DWM and the hybrid policies). Nonetheless, this gap can be well characterized. Specifically, in the G-FBS policy, an L-frame contains at most $H\sqrt{n}$ packets that are not served whenever there is a success. Further, these (at most) $H\sqrt{n}$ leftover packets will be among the oldest $f(n)$ packets (in the S-frame) in the next time-slot for large n, due to our choice of $f(n) \in \omega(\sqrt{n})$. Hence, another success will serve all the leftover packets. This implies that at most $x+1$ successes are needed to completely serve x frames, for any finite integer $x > 0$. *In fact, this property is the key reason for a one-time-slot shift in the guaranteed rate-function by G-FBS, which leads to the near-optimal delay rate-function*, as we show in the following theorem.

Theorem 4: Under Assumptions 2 and 3, G-FBS policy achieves a near-optimal rate-function, as given in Eq. (3).

The proof of Theorem 4 follows a similar line of argument as in the proof for rate-function delay optimality of FBS (Theorem 2 in [13]). We consider all the events that lead to the delay-violation event $\{W(0) > b\}$, which can be caused by two factors: bursty arrivals and sluggish service. On the one hand, if there are a large number of arrivals in certain period, say of length t time-slots, which exceeds the maximum number of packets that can be served in a period of $t+b+1$ time-slots, then it unavoidably leads to a delay-violation. On the other hand, suppose that there is at least one packet arrival at certain time, and that under G-FBS, a success does not occur in any of the following $b+1$ time-slots (including the time-slot when the packet arrives), then it also leads to a delay-violation. Each of these two possibilities has a corresponding rate-function for its probability of occurring. Large-deviations theory then tells us that the rate-function for delay-violation is determined by the smallest rate-function among these possibilities (i.e., "rare events occur in the most likely way"). We can then show that $I(b) \geq I_U(b-1) \geq I^*(b-1)$ for any integer $b > 0$, where $I(\cdot)$ is the rate-function attained by G-FBS, $I_U(\cdot)$ is the upper bound that we derived in Section III, and $I^*(\cdot)$ is the optimal rate-function, respectively. We provide the detailed proof of Theorem 4 in Appendix D.

Remark: Note that the gap between the optimal rate-function and the above near-optimal rate-function is likely to be quite small. For example, in the case where the arrival is either 1 or 0, the near-optimal rate-function implies $I(b) \geq \frac{b}{b+1} I^*(b)$, since we have $I^*(b) = (b+1)\log\frac{1}{1-q}$ for this case [15].

Finally, we make use of the following dominance property of D-SSG over G-FBS.

Lemma 2: For any given sample path and for any value of h, by the end of any time-slot t, D-SSG has served every packet that G-FBS has served.

We prove Lemma 2 by contradiction. The proof follows a similar argument as in the proof of Lemma 7 in [13],

and is provided in our online technical report [24]. Then, the near-optimal rate-function of D-SSG (Theorem 3) follows immediately from Lemma 2 and Theorem 4.

Remark: Note that D-SSG combined with DWM-n policy, can also yield an $O(n^{2.5}\log n)$-complexity hybrid policy that is both throughput-optimal and rate-function delay-optimal. We omit the details since the treatment follows similarly as that for hybrid DWM-n-MWS policy [15].

So far, we have shown that our proposed low-complexity D-SSG policy achieves both throughput optimality and near-optimal delay rate-function. In the next section, we will show through simulations that in all scenarios we consider, *D-SSG not only exhibits a near-optimal delay rate-function, but also empirically has a similar delay performance to the rate-function delay-optimal policies such as DWM and the hybrid DWM-n-MWS policy.*

V. Simulation Results

In this section, we conduct simulations to compare scheduling performance of our proposed D-SSG policy with DWM, hybrid DWM-n-MWS (called Hybrid for short), D-MWS, and Q-SSG. We simulate these policies in Java and compare the empirical probabilities that the largest HOL delay in the system in any given time-slot exceeds an integer threshold b, i.e., $\mathbb{P}(W(0) > b)$.

Same as in [15], we consider bursty arrivals that are driven by a two-state Markov chain and that are correlated over time. (We obtained similar results for *i.i.d.* 0-L arrivals, and omit them here.) For each user, there are 5 packet-arrivals when the Markov chain is in state 1, and there is no arrivals when it is in state 2. The transition probability of the Markov chain is given by the matrix $[0.5, 0.5; 0.1, 0.9]$, and the state transitions occur at the end of each time-slot. The arrivals for each user are correlated over time, but they are independent across users. For the channel model, we first assume *i.i.d.* ON-OFF channels with unit capacity, and set $q = 0.75$. We later consider more general scenarios with heterogeneous users and bursty channels that are correlated over time. We run simulations for a system with n servers and n users, where $n \in \{10, 20, \ldots, 100\}$. The simulation period lasts for 10^7 time-slots for each policy and each system.

The results are summarized in Fig. 2, where the complexity of each policy is also labeled. In order to compare the rate-function $I(b)$ as defined in Eq. (2), we plot the probability over the number of channels or users, i.e., n, for a fixed value of threshold b. The negative of the slopes of the curves can be viewed as the rate-function for each policy. In Fig. 2, we report the results only for $b = 4$, and the results are similar for other values of threshold b. From Fig. 2, we observe that D-SSG has a similar delay performance to that of DWM and Hybrid, which are both known to be rate-function delay-optimal. This not only supports our theoretical results that D-SSG guarantees a near-optimal rate-function, but also implies that D-SSG empirically performs very well while enjoying a lower complexity. Further, we observe that D-SSG consistently outperforms its queue-length-based counterpart, Q-SSG, despite the fact that in [9], it has been shown through simulations that Q-SSG *empirically* achieves near-optimal queue-length performance. This provides a further evidence that good queue-length performance does not necessarily translate into good delay performance. The results also show that D-MWS yields a zero rate-function, as expected.

We also plot the probability for delay threshold b as in [5], [9], [10], [12], [13], [15] to investigate the performance of different policies for fixed n. In Fig. 3, we report the results for $n = 10$, and the results are similar for other values of n. From Fig. 3, we observe that D-SSG consistently performs closely to DWM and Hybrid for almost all values of b that we consider. We also observe that D-SSG consistently outperforms its queue-length-based counterpart, Q-SSG.

In addition, we compute the average time required for the operations of each policy within one scheduling cycle, when $n = 100$. Running simulations in a PC with Intel Core i7-2600 3.4GHz CPU and 8GB memory, D-SSG requires roughly 0.3 millisecond to finish all of the required operations within one scheduling cycle (which, for example, is 1 millisecond in LTE systems), while the two-stage Hybrid policy needs 7-10 times more. This, along with the above simulation results, implies that in practice D-SSG is more suitable for actual implementations than the hybrid policies, although D-SSG does not guarantee rate-function delay optimality.

Further, we evaluate scheduling performance of different policies in more realistic scenarios, where users are *heterogeneous* and channels are *correlated over time*. Specifically, we consider channels that can be modeled as a two-state Markov chain, where the channel is "ON" when the Markov chain is in state 1, and is "OFF" when it is in state 2. We assume that there are two classes of users: users with an odd index are called *near-users*, and users with an even index are called *far-users*. Different classes of users see different channel conditions: near-users see better channel condition, and far-users see worse channel condition. We assume that the transition probability matrices of channels for near-users and far-users are $[0.833, 0.167; 0.5, 0.5]$ and $[0.5, 0.5; 0.167, 0.833]$, respectively. The arrival processes are assumed to be the same as in the previous case.

The results are summarized in Fig. 4. We observe similar results as in the previous case with homogeneous users and *i.i.d.* channels in time. In particular, D-SSG exhibits a rate-function that is similar to that of DWM and Hybrid, although its delay performance is slightly worse. Note that in this scenario, a rate-function delay-optimal policy is *not* known yet. Hence, for future work, it would be interesting to understand how to design rate-function delay-optimal or near-optimal policies in general scenarios.

VI. Conclusion

In this paper, we developed a practical and low-complexity greedy scheduling policy (D-SSG) that not only achieves throughput optimality, but also guarantees a near-optimal delay rate-function, for multi-channel wireless networks. Our studies reveal that throughput optimality is relatively easier to achieve in such multi-channel systems, while there exists an explicit trade-off between complexity and delay performance. If one

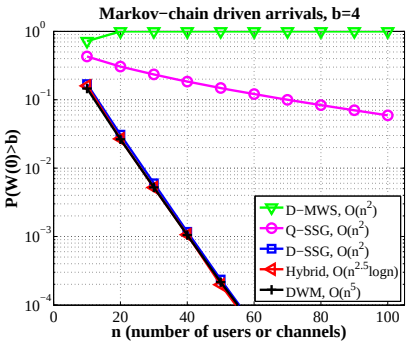


Fig. 2. Performance comparison of different scheduling policies in the case with homogeneous *i.i.d.* channels, for delay threshold $b = 4$.

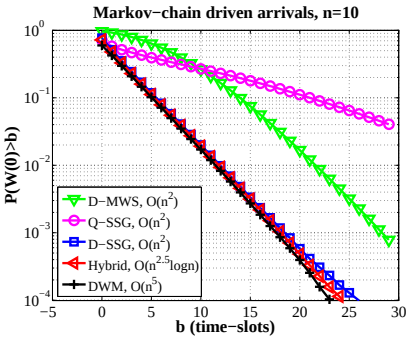


Fig. 3. Performance comparison of different scheduling policies in the case with homogeneous *i.i.d.* channels, for $n = 10$ channels or users.

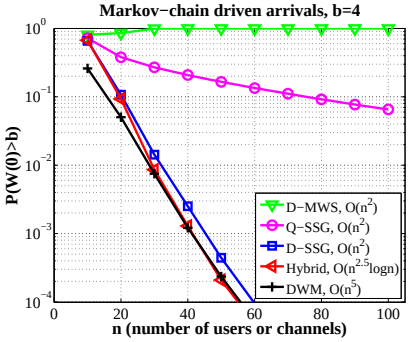


Fig. 4. Performance comparison of different scheduling policies in the case with Markov-chain driven heterogeneous channels, for delay threshold $b = 4$.

can bear a minimal drop in the delay performance, lower-complexity scheduling policies can be exploited.

The analytical results in this paper are derived for the *i.i.d.* ON-OFF channel model with unit channel capacity. An interesting direction for future work is to study general multi-rate channels that can be correlated over time. We note that this problem will become much more challenging. For example, even for an *i.i.d.* 0-K channel model with channel capacity $K > 1$, it is still unclear whether there exists a scheduling policy that can guarantee both optimal throughput and optimal/near-optimal asymptotic delay performance. Another direction for future work is to consider heterogeneous users with different arrival processes and different delay requirements. In these more general scenarios, it may be worth exploring how to find efficient schedulers that can guarantee a nontrivial lower bound of the optimal rate-function, if it turns out to be too difficult to achieve or prove the optimal asymptotic delay performance itself. Nonetheless, we believe that the results derived in this paper will provide useful insights for designing high-performance scheduling policies for more general scenarios.

Appendix A
Proof of Theorem 1

We consider event $\mathcal{E}_1$ and a sequence of events $\mathcal{E}_2^c$ implying the occurrence of event $\{W(0) > b\}$.

Event $\mathcal{E}_1$: Suppose that there is a packet that arrives to the network in time-slot $-b-1$. Without loss of generality, we assume that the packet arrives to queue Q_1. Further, suppose that Q_1 is disconnected from all the n servers in all the time-slots from $-b-1$ to -1.

Then, at the beginning of time-slot 0, this packet is still in the network and has a delay of $b+1$. This implies $\mathcal{E}_1 \subseteq \{W(0) > b\}$. Note that the probability that event $\mathcal{E}_1$ occurs can be computed as

$$\mathbb{P}(\mathcal{E}_1) = (1-q)^{n(b+1)} = e^{-n(b+1)I_X}.$$

Hence, we have

$$\mathbb{P}(W(0) > b) \geq e^{-n(b+1)I_X},$$

and thus

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq (b+1)I_X.$$

Event $\mathcal{E}_2^c$: Consider any fixed $c \in \{0, 1, \ldots, b\}$. Fix any $\epsilon > 0$, and choose t such that $I_{AG}(t, b-c) \leq I_{AG}(b-c) + \epsilon$. Suppose that from time-slot $-t-b$ to $-b-1$, the total number of packet arrivals to the system is greater than $nt + n(b-c)$, and let $p_{(b-c)}$ denote the probability that this event occurs. Then, from the definitions of $I_{AG}(t,x)$ and $I_{AG}(t)$, we know

$$\liminf_{n\to\infty} \frac{-1}{n} \log p_{(b-c)} = I_{AG}(t, b-c) \leq I_{AG}(b-c) + \epsilon.$$

Clearly, the total number of packets that are served in any time-slot is no greater than n. Hence, at the end of time-slot $-b-1$, there are at least $n(b-c)+1$ packets remaining in the system. Moreover, at the end of time-slot $-c-1$, the system contains at least one packet that arrived before time-slot $-b$. Without loss of generality, we assume that this packet is in Q_1. Now, assume that Q_1 is disconnected from all the n servers in the next c time-slots, i.e., from time-slot $-c$ to -1. This occurs with probability $(1-q)^{cn} = e^{-ncI_X}$, independently of all the past history. Hence, at the beginning of time-slot 0, there is still a packet that arrived before time-slot $-b$. Hence, we have $W(0) > b$ in this case. This implies $\mathcal{E}_2^c \subseteq \{W(0) > b\}$. Note that the probability that event $\mathcal{E}_2^c$ occurs can be computed as

$$\mathbb{P}(\mathcal{E}_2^c) = p_{(b-c)} e^{-ncI_X}.$$

Hence, we have

$$\mathbb{P}(W(0) > b) \geq p_{(b-c)} e^{-ncI_X},$$

and thus

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq I_{AG}(b-c) + \epsilon + cI_X.$$

Since the above inequality holds for any $c \in \{0, 1, \ldots, b\}$ and all $\epsilon > 0$, by letting ϵ tend to 0 and taking the minimum over all $c \in \{0, 1, \ldots, b\}$, we have

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq \min_{c\in\{0,1,\ldots,b\}} \{I_{AG}(b-c) + cI_X\}.$$

Considering both event $\mathcal{E}_1$ and events $\mathcal{E}_2^c$, we have

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq \min\{\min_{c\in\{0,1,\ldots,b\}} \{I_{AG}(b-c) + cI_X\}, (b+1)I_X\}.$$

APPENDIX B
DELAY-BASED QUEUE-SIDE-GREEDY (D-QSG) AND SAMPLE-PATH EQUIVALENCE

Delay-based Queue-Side-Greedy (D-QSG) policy, in an iterative manner too, schedules the oldest packets in the system one-by-one whenever possible. In this sense, D-QSG can be viewed as an intuitive approximation of the Oldest Packet First (OPF) policies[3] [15] that have been shown to be rate-function delay-optimal. We later prove an important sample-path equivalence result (Lemma 3) that will be used in proving Lemma 1 and thus the main result of this paper (Theorem 3).

We start by presenting some additional notations. In the D-QSG policy, there are at most n rounds in each time-slot t. By slightly abusing the notations, we let $Q_i^k(t)$, $Z_{i,l}^k(t)$ and $W_i^k(t) = Z_{i,1}^k(t)$ denote the length of queue Q_i, the delay of the l-th packet of Q_i, and the HOL delay of Q_i after the k-th round in time-slot t under D-QSG, respectively. Let $\Upsilon_k(t)$ denote the set of indices of the available servers at the beginning of the k-th round, and let $\Psi_k(t)$ denote the set of queues that have the largest HOL delay among all the queues that are connected to at least one server in $\Upsilon_k(t)$ at the beginning of the k-th round, i.e., $\Psi_k(t) \triangleq \{1 \le i \le n \mid W_i^{k-1}(t) \cdot \mathbb{1}_{\{\sum_{j\in\Upsilon_k(t)} C_{i,j}(t)>0\}} = \max_{1\le l\le n} W_l^{k-1}(t) \cdot \mathbb{1}_{\{\sum_{j\in\Upsilon_k(t)} C_{l,j}(t)>0\}}\}$. Also, let $i(k,t)$ be the index of the queue that is served in the k-th round of time-slot t, and let $j(k,t)$ be the index of the server that serves $Q_{i(k,t)}$ in that round. We then specify the operations of D-QSG as follows.

Delay-based Queue-Side-Greedy (D-QSG) policy: In each time-slot t,

1) Initialize $k = 1$ and $\Upsilon_1 = \{1, 2, \ldots, n\}$.
2) In the k-th round, allocate server $S_{j(k,t)}$ to $Q_{i(k,t)}$, where

$$i(k,t) = \min\{i \mid i \in \Psi_k(t)\},$$
$$j(k,t) = \min\{j \in \Upsilon_k(t) \mid C_{i(k,t),j}(t) = 1\}.$$

That is, in the k-th round, we consider the queues that have the largest HOL delay among those that have at least one available server connected (i.e., the queues in set $\Psi_k(t)$), and break ties by picking the queue with the smallest index (i.e., $Q_{i(k,t)}$). We then choose an available server that is connected to queue $Q_{i(k,t)}$, and break ties by picking the server with the smallest index (i.e., server $S_{j(k,t)}$), to serve $Q_{i(k,t)}$. At the end of the k-th round, update the length of $Q_{i(k,t)}$ to account for service, i.e., set $Q_{i(k,t)}^k(t) = \left(Q_{i(k,t)}^{k-1}(t) - C_{i(k,t),j(k,t)}(t)\right)^+$ and $Q_i^k(t) = Q_i^{k-1}(t)$ for all $i \neq i(k,t)$. Also, update the HOL delay of $Q_{i(k,t)}$, by setting $W_{i(k,t)}^k(t) = Z_{i(k,t),1}^k(t) = Z_{i(k,t),2}^{k-1}(t)$ if $Q_{i(k,t)}^k(t) > 0$, and $W_{i(k,t)}^k(t) = 0$ otherwise, and setting $W_i^k(t) = W_i^{k-1}(t)$ for all $i \neq i(k,t)$.
3) Stop if k equals n. Otherwise, increase k by 1, set $\Upsilon_k(t) = \Upsilon_{k-1}(t)\backslash\{j(k,t)\}$, and repeat step 2.

Remark: Note that D-QSG is only used for assisting the rate-function delay analysis of D-SSG and may not be suitable for practical implementation due to its $O(n^3)$ complexity. This is because there are at most n rounds, and in each round, it takes $O(n^2 + n) = O(n^2)$ time to find a queue that has at least one connected and available server (which takes $O(n^2)$ time to check for all queues) and that has the largest HOL delay (which takes $O(n)$ time to compare).

[3]A scheduling policy $\mathbf{P}$ is said to be an OPF policy if in any time-slot, policy $\mathbf{P}$ can serve the k oldest packets in that time-slot for the largest possible value of $k \in \{1, 2, \ldots, n\}$.

The following lemma states the sample-path equivalence property between D-QSG and D-SSG under the tie-breaking rules specified in this paper.

Lemma 3: For the same sample path, i.e., same realizations of arrivals and channel connectivity, D-QSG and D-SSG pick the same schedule in every time-slot.

Proof: We prove Lemma 3 by induction. It suffices to prove that for any given system, i.e., for any given set of packets after arrivals and for any channel realizations, both D-SSG and D-QSG pick the same schedule. Suppose that there are K packets in the system. Let x_k denote the k-th oldest packet in the system. We want to show that packet x_k is either served by the same server under both D-SSG and D-QSG, or is not served by any server under both D-SSG and D-QSG. We denote the set of the k oldest packets by $\mathcal{P}_k \triangleq \{x_r \mid r \le k\}$, and denote the set of the first k servers by $\mathcal{S}_k \triangleq \{S_j \mid j \le k\}$. Let $S_{j(r)}$ denote the server allocated to serve the r-th oldest packet under D-QSG. We prove it by induction method.

Base case: Consider packet x_1, i.e., the oldest packet, and consider two cases: under D-QSG, 1) packet x_1 is served by $S_{j(1)}$; 2) packet x_1 is not served by any server.

In Case 1), we want to show that packet x_1 is also served by the same server $S_{j(1)}$ under D-SSG. Note that packet x_1 is the oldest packet in the system and is the first packet to be considered under D-QSG. Since it is served by $S_{j(1)}$, from the tie-breaking rule of D-QSG, we know that the queue that contains packet x_1 is disconnected from all the servers in set $\mathcal{S}_{j(1)}$ except server $S_{j(1)}$. Now, we consider the server allocation under D-SSG, which allocates servers one-by-one in an increasing order of the server index. Since all the servers in set $\mathcal{S}_{j(1)}$ except for server $S_{j(1)}$ are disconnected from the queue containing packet x_1, these servers cannot be allocated to packet x_1 in the first $j(1)-1$ rounds under D-SSG. While in the $j(1)$-th round, D-SSG must allocate server $S_{j(1)}$ to packet x_1, since the queue that contains packet x_1 is the queue that has the largest HOL among the queues that are connected to server $S_{j(1)}$.

In Case 2), packet x_1 is the first packet to be considered under D-QSG, but is not served by any server. This implies that no servers are connected to the queue that contains packet x_1. Hence, packet x_1 cannot be served under D-SSG either.

Combining the above two cases, we prove the base case.

Induction step: Consider an integer $k \in \{1, 2, \ldots, K-1\}$. Suppose that every packet in set $\mathcal{P}_k$ is either served by the same server under both D-QSG and D-SSG, or is not served by any server under both D-QSG and D-SSG. We want to show that this also holds for every packet in set $\mathcal{P}_{k+1}$. Clearly, it suffices to consider only packet x_{k+1} (i.e., the $(k+1)$-th oldest packet in the system), as the other packets all satisfy the condition from the induction hypothesis. We next consider two cases: under D-QSG, 1) packet x_{k+1} is scheduled by a server under D-QSG; 2) x_{k+1} is not served by any server.

In Case 1), suppose that packet x_{k+1} is served by server $S_{j(k+1)}$ under D-QSG. We want to show that packet x_{k+1} is also served by server $S_{j(k+1)}$ under D-SSG. We first show that under D-SSG, packet x_{k+1} cannot be served in the first $j(k+1)-1$ rounds. Note that under D-QSG, packet x_{k+1} is served by server $S_{j(k+1)}$. This implies that any server in set $\mathcal{S}_{j(k+1)-1}$ is either disconnected from the queue that contains packet x_{k+1} or has already been allocated to packets in set $\mathcal{P}_k$ under D-QSG. This, along with the induction hypothesis, further implies that under D-SSG, in the first $j(k+1)-1$ rounds, the servers under consideration are either disconnected from the queue that contains packet x_{k+1} or allocated to packets in set $\mathcal{P}_k$. Hence, packet x_{k+1} cannot be scheduled in the first $j(k+1)-1$ rounds under D-SSG. Next, we want to show that packet x_{k+1} must be served by server $S_{j(k+1)}$ in the $j(k+1)$-th round under D-SSG. Let $\mathcal{P}'_k \subseteq \mathcal{P}_k$ denote the set of packets among the k oldest packets that are not served under both D-QSG and D-SSG. Then, all the queues that contain packets in set $\mathcal{P}'_k$ must be disconnected from server $S_{j(k+1)}$, otherwise some packet $x_r \in \mathcal{P}'_k$ should be served by server $S_{j(k+1)}$ under D-QSG. On the other hand, the induction hypothesis implies that any packet $x_r \in \mathcal{P}_k \backslash \mathcal{P}'_k$ must be served by some server $S_{j(r)}$, under D-SSG, where $j(r) \neq j(k+1)$. Hence, D-SSG does not allocate server $S_{j(k+1)}$ to any packet in set $\mathcal{P}_k$. Therefore, in the $(k+1)$-th round, D-SSG must allocate server $S_{j(k+1)}$ to packet x_{k+1}, since the queue that contains packet x_{k+1} has the largest HOL delay among the queues that are connected to server $S_{j(k+1)}$.

In Case 2), packet x_{k+1} is not served by any server under D-QSG. This implies that the queue that contains packet x_{k+1} is disconnected from all the servers in set $\mathcal{S}_n \backslash \{S_{j(r)} \mid r \in \mathcal{P}_k \backslash \mathcal{P}'_k\}$, i.e., the set of available servers when considering packet x_{k+1}. On the other hand, the induction hypothesis implies that under D-SSG, all the servers in set $\{S_{j(r)} \mid r \in \mathcal{P}_k \backslash \mathcal{P}'_k\}$ are also allocated to packets in set $\mathcal{P}_k \backslash \mathcal{P}'_k$. Hence, packet x_{k+1} cannot be served by any server under D-SSG either.

Combining the above two cases, we prove the induction step. This completes the proof. ∎

Note that under D-SSG, in each round, when a server has multiple connected queues that have the largest HOL delay, we break ties by picking the queue with the smallest index. Presumably, one can take other arbitrary tie-breaking rules. However, it turns out to be much more difficult to directly analyze the rate-function performance for a greedy policy from the server side (like D-SSG) without using the above equivalence property. For example, as we mentioned earlier, the authors of [9], [10] were only able to prove a positive (queue-length) rate-function for Q-SSG in more restricted scenarios. Hence, our choice of the above simple tie-breaking rule is in fact quite important for proving the above sample-path equivalence result, which in turn plays a critical role in proving a key property of D-SSG (Lemma 1) and thus near-optimal rate-function of D-SSG (Theorem 3). Nevertheless, we would expect that one can choose arbitrary tie-breaking rules for D-SSG in practice.

APPENDIX C
PROOF OF LEMMA 1

We then divide the proof into two parts (Lemmas 4 and 5).

Lemma 4: Consider a set of n packets. Consider any function $f(n) < \frac{n}{2}$, which is strictly increasing with n. The D-SSG policy is applied to schedule these n packets. Then, there exists a finite integer $N_{X1} > 0$ such that for all $n \geq N_{X1}$, with probability no smaller than $1-(1-q)^{n-f(n)\log^2 n}$, D-SSG schedules all the oldest $f(n)$ packets among the n packets.

Proof: Since Lemma 4 is focused on the oldest $f(n)$ packets in the set, it is easier to consider the D-QSG policy instead, which in an iterative manner schedules the oldest packets first. Due to the sample-path equivalence between D-SSG and D-QSG (Lemma 3), it is sufficient to prove that the result of Lemma 4 holds for D-QSG.

Suppose that the oldest $f(n)$ packets among the n packets are from k different queues, where $k \leq f(n)$. It is easy to see that if each of the k queues is connected to no less than $f(n)$ servers, then all of these oldest $f(n)$ packets will be served. Specifically, because D-QSG gives a higher priority to an older packet, the above condition guarantees that when D-QSG schedules any of the oldest $f(n)$ packets, there will always be at least one available server that is connected to the queue containing this packet.

Now, consider any queue Q_i. We want to compute the probability that Q_i is connected to no less than $f(n)$ servers. We first compute the probability that Q_i is connected to less than $f(n)$ servers:

$$\begin{aligned}
&\mathbb{P}(Q_i \text{ is connected to less than } f(n) \text{ servers})\\
=&\textstyle\sum_{j=0}^{f(n)-1} \mathbb{P}(Q_i \text{ is connected to } j \text{ servers})\\
=&\textstyle\sum_{j=0}^{f(n)-1} \binom{n}{j} q^j (1-q)^{n-j}\\
\leq& f(n) n^{f(n)} (1-q)^{n-f(n)}.
\end{aligned}$$

Next, choose N_{X1} such that $f(n)n^{f(n)} < (\frac{1}{1-q})^{n-f(n)}$ and $f^2(n)n^{f(n)} \leq (\frac{1}{1-q})^{f(n)(\log^2 n-1)}$ for all $n \geq N_{X1}$. Such an N_{X1} exists because $\log(f(n)n^{f(n)}) = \Theta(f(n)\log n)$ and $\log((\frac{1}{1-q})^{n-f(n)}) = \Theta(n)$, hence, we have $\log(f(n)n^{f(n)}) < \log((\frac{1}{1-q})^{n-f(n)})$ and thus $f(n)n^{f(n)} < (\frac{1}{1-q})^{n-f(n)}$ for large enough n; and similarly, because $\log(f^2(n)n^{f(n)}) = \Theta(f(n)\log n)$ and $\log((\frac{1}{1-q})^{f(n)(\log^2 n-1)}) = \Theta(f(n)\log^2 n)$, hence, we have $\log(f^2(n)n^{f(n)}) \leq \log((\frac{1}{1-q})^{f(n)(\log^2 n-1)})$ and thus $f^2(n)n^{f(n)} \leq (\frac{1}{1-q})^{f(n)(\log^2 n-1)}$ for large enough n. Then, the probability that each of the k queues is connected to no less than $f(n)$ servers is:

$$\begin{aligned}
&\mathbb{P}(\text{Each of the } k \text{ queues is connected to no less than } f(n) \text{ servers})\\
\geq&(1-f(n)n^{f(n)}(1-q)^{n-f(n)})^k\\
\overset{(a)}{\geq}&(1-f(n)n^{f(n)}(1-q)^{n-f(n)})^{f(n)}\\
\overset{(b)}{\geq}&1-f^2(n)n^{f(n)}(1-q)^{n-f(n)}\\
\overset{(c)}{\geq}&1-(1-q)^{n-f(n)\log^2 n}
\end{aligned}$$

for all $n \geq N_{X1}$, where (a) is from our choice of N_{X1} and the fact that $k \leq f(n)$, (b) is from our choice of N_{X1} and

Bernoulli's inequality (i.e., $(1+x)^r \geq 1+rx$ for every real number $x \geq -1$ and every integer $r \geq 0$), and (c) is from our choice of N_{X1}. This completes the proof. ■

Lemma 5: Consider a set of n packets satisfying that no more than $2H$ packets are from the same queue, where $H > 4$ is any integer constant independent of n. The D-SSG policy is applied to schedule these n packets. Then, there exists a finite integer $N_{X2} > 0$ such that for all $n \geq N_{X2}$, with probability no smaller than $1-(1-q)^n$, D-SSG schedules at least $n - H\sqrt{n}$ packets among the n packets.

Proof: Consider the D-SSG policy. We first compute the probability that some $H\sqrt{n}$ packets are not scheduled by D-SSG, which is equivalent to the event that some $H\sqrt{n}$ servers are not allocated to any packet by D-SSG.

Consider any arbitrary set of servers $\Xi = \{S_{r_j} \mid j = 1, 2, \ldots, H\sqrt{n}\}$, where $r_i < r_j$ if $i < j$. Clearly, we have $r_j \leq n - H\sqrt{n} + j$ for all $j \in \{1, 2, \ldots, H\sqrt{n}\}$. Consider the r_j-th server S_{r_j}. Then, the number of remaining packets is at least $n - r_j + 1$ at the beginning of the r_j-th round. Since no more than $2H$ packets are from the same queue, there are at least $\lceil \frac{n-r_j+1}{2H} \rceil$ queues that are non-empty at the beginning of the r_j-th round. Then, the probability that server S_{r_j} is not allocated to any packet is no greater than $(1-q)^{\lceil \frac{n-r_j+1}{2H} \rceil} \leq (1-q)^{\frac{n-r_j+1}{2H}}$. Hence,

$$\begin{aligned}
&\mathbb{P}(\text{None of the servers in a given set } \Xi \text{ is allocated}) \\
\leq &\textstyle\prod_{j=1}^{H\sqrt{n}} (1-q)^{\frac{n-r_j+1}{2H}} \\
\leq &\textstyle\prod_{j=1}^{H\sqrt{n}} (1-q)^{\frac{n-(n-H\sqrt{n}+j)+1}{2H}} \\
\leq &(1-q)^{\frac{1}{2H}(1+2+\cdots+H\sqrt{n})} \\
= &(1-q)^{\frac{H}{4}n+\frac{\sqrt{n}}{4}}
\end{aligned}$$

Since $H > 4$, there exists an N_{X2} such that $n^{H\sqrt{n}}(1-q)^{\frac{H}{4}n+\frac{\sqrt{n}}{4}} \leq (1-q)^n$ for all $n \geq N_{X2}$. Such an N_{X2} exists because $\log(n^{H\sqrt{n}}) = \Theta(\sqrt{n}\log n)$ and $\log((\frac{1}{1-q})^{\frac{H}{4}n+\frac{\sqrt{n}}{4}-n}) = \Theta(n^{\frac{H}{4}-1})$, hence, $\log(n^{H\sqrt{n}}) \leq \log((\frac{1}{1-q})^{\frac{H}{4}n+\frac{\sqrt{n}}{4}-n})$ and thus $n^{H\sqrt{n}}(1-q)^{\frac{H}{4}n+\frac{\sqrt{n}}{4}} \leq (1-q)^n$ for large enough n. Then, we can compute the probability that some $H\sqrt{n}$ servers are not allocated as

$$\begin{aligned}
&\mathbb{P}(\text{Some } H\sqrt{n} \text{ servers are not allocated}) \\
\leq &\binom{n}{H\sqrt{n}} \mathbb{P}(\text{None of the servers in a given set } \Xi \text{ is allocated}) \\
\leq &n^{H\sqrt{n}}(1-q)^{\frac{H}{4}n+\frac{\sqrt{n}}{4}} \\
\leq &(1-q)^n
\end{aligned}$$

for all $n \geq N_{X2}$, where the last inequality is due to our choice of N_{X2}.

Therefore, we have

$$\begin{aligned}
&\mathbb{P}(\text{At least } n - H\sqrt{n} \text{ packets are scheduled}) \\
= &1 - \mathbb{P}(\text{Less than } n - H\sqrt{n} \text{ packets are scheduled}) \\
= &1 - \mathbb{P}(\text{Greater than } H\sqrt{n} \text{ packets are not scheduled}) \\
\geq &1 - \mathbb{P}(\text{At least } H\sqrt{n} \text{ packets are not scheduled}) \\
= &1 - \mathbb{P}(\text{Some } H\sqrt{n} \text{ packets are not scheduled}) \\
= &1 - \mathbb{P}(\text{Some } H\sqrt{n} \text{ servers are not allocated}) \\
\geq &1 - (1-q)^n,
\end{aligned}$$

for all $n \geq N_{X2}$. ■

By applying Lemmas 4 and 5, and choosing $N_X \triangleq \max\{N_{X1}, N_{X2}, N_{X3}\}$, where N_{X3} is such that $n - H\sqrt{n} > f(n)$ for all $n \geq N_{X3}$, we show that for all $n \geq N_X$, with probability no smaller than $1 - 2(1-q)^{n-f(n)\log^2 n}$, D-SSG schedules at least $n - H\sqrt{n}$ packets including the oldest $f(n)$ packets among the n packets.

Appendix D
Proof of Theorem 4

The proof follows a similar argument for the proof of Theorem 2 in [13].

We start by defining $I_0 \triangleq I_U(b-1) = \min\{bI_X, \min_{0\leq c\leq b-1}\{I_{AG}(b-1-c) + cI_X\}\} \geq I^*(b-1)$. Consider any fixed $\epsilon > 0$, and define $I_0^\epsilon \triangleq \min\{bI_X, \min_{0\leq c\leq b-1}\{I_{AG}(b-1-c) - \epsilon + cI_X\}\}$. Then, we have $\lim_{\epsilon\to 0} I_0^\epsilon = I_0$.

We then choose the value of parameter h for G-FBS based on the statistics of the arrival process. We fix $\delta < \frac{2}{3}$ and $\eta < \frac{p}{2}$. Then, from Assumption 3, there exists a positive function $I_B(\eta,\delta)$ such that for all $n \geq N_B(\eta,\delta)$ and $t \geq T_B(\eta,\delta)$, we have

$$\mathbb{P}\left(\frac{\sum_{\tau=r+1}^{r+t} \mathbb{1}_{\{|A(\tau)-pn|>\eta n\}}}{t} > \delta\right) < \exp(-ntI_B(\eta,\delta)),$$

for any integer r. We then choose

$$h = \max\left\{T_B(\eta,\delta), \left\lceil \frac{1}{(p-\eta)(1-\frac{3\delta}{2})} \right\rceil, \left\lceil \frac{2I_0^\epsilon}{I_B(\eta,\delta)} \right\rceil, 4\right\} + 1.$$

The reason for choosing the above value of h will become clear later on. Recall from Assumption 2 that L is the maximum number of packets that can arrive to a queue in any time-slot t. Then, $H = Lh$ is the maximum number of packets that can arrive to a queue during an interval of h time-slots, and is thus the maximum number of packets from the same queue in a frame. This also implies that in the S-frame, no more than $2H$ packets are from the same queue.

Next, we define the following notions associated with the G-FBS policy. Let $F(t)$ denote the number of unserved frames in time-slot t, and let $R(t)$ denote the remaining available space (where the unit is packet) in the end-of-line frame at the end of time-slot t. Also, let $X_F(t)$ denote the indicator function of whether a success occurs in time-slot t. That is, $X_F(t) = 1$ if there is a success, and $X_F(t) = 0$ otherwise. Recall that $n_0 = n - H\sqrt{n}$. Then, we can write a recursive equation for $F(t)$:

$$F(t) = (F(t-1) + \left\lceil \frac{A(t) - R(t-1)}{n_0} \right\rceil - X_F(t), 0)^+, \quad (4)$$

$$R(t) = \mathbb{1}_{\{F(t)>0\}} \cdot ((R(t-1) - A(t)) \mod n_0). \quad (5)$$

Let $M(t) \leq H\sqrt{n}$ denote the number of packets in the L-frame at the beginning of time-slot t, and let $P(t) \leq n_0$ denote the number of packets in the HOL frame at the beginning of time-slot t. Then, at the beginning of time-slot t, the number of packets in the S-frame is equal to $M(t)+P(t)$. Let $D(t) \leq M(t) + P(t)$ denote the number of packets served from the

S-frame if a success occurs in time-slot t. Then, we have the following recursive equation for $M(t)$:

$$M(t+1) = \begin{cases} M(t) + P(t) - D(t), & \text{if } X_F(t) = 1, \\ M(t), & \text{otherwise.} \end{cases}$$

Also, we let

$$X_F(t_1, t_2) = \sum_{\tau=t_1}^{t_2} X_F(\tau) \mathbb{1}_{\{\{F(\tau)>0\} \cup \{M(\tau)>0\}\}}$$

denote the the total number of successes in the interval from time-slot t_1 to t_2 when the S-frame is non-empty (i.e., the number of unserved frames is greater than zero or the L-frame is non-empty).

Note that the arriving time of a frame is the time when its first packet arrives. Let $R_0 = R(t_1 - 1)$ denote the available space in the end-of-line frame at the end of time-slot $t_1 - 1$. Then, we let $A_F^{R_0}(t_1, t_2)$ denote the number of new frames that arrive from time-slot t_1 to t_2. When $R_0 = 0$, we use $A_F(t_1, t_2)$ to denote $A_F^{R_0}(t_1, t_2)$ for notational convenience.

Let $L(-b)$ be the last time before $-b$, when the number of unserved frames is equal to zero. Then, given that $L(-b) = -t - b - 1$, where $t > 0$, the number of unserved frames never becomes zero during interval $[-t-b, -b-1]$. Let $U(0)$ denote the indicator function of whether at time-slot 0 the L-frame contains a packet that arrives before time-slot $-b$, i.e., $U(0) = 1$ if at time-slot 0 the L-frame contains a packet that arrives before time-slot $-b$, and $U(0) = 0$, otherwise. Let $\mathcal{E}_t^{\alpha_1}$ denote the event that the number of frames that arrive during interval $[-t-b, -b-1]$ is greater than the total number of successes during interval $[-t-b, -1]$ when the S-frame is non-empty, i.e.,

$$\mathcal{E}_t^{\alpha_1} = \{A_F(-t-b, -b-1) > X_F(-t-b, -1)\}.$$

Let $\mathcal{E}_t^{\alpha_2}$ denote the event that the number of frames that arrive during interval $[-t-b, -b-1]$ is equal to the total number of successes during interval $[-t-b, -1]$ when the S-frame is non-empty, and at time-slot 0 the L-frame contains a packet that arrives before time-slot $-b$, i.e.,

$$\mathcal{E}_t^{\alpha_2} = \{A_F(-t-b, -b-1) = X_F(-t-b, -1), U(0) = 1\}.$$

Letting $\mathcal{E}_t^{\alpha} = \mathcal{E}_t^{\alpha_1} \cup \mathcal{E}_t^{\alpha_2}$, we have

$$\begin{aligned} &\{L(-b) = -t-b-1, W(0) > b\} \\ &\quad = \{L(-b) = -t-b-1, \mathcal{E}_t^{\alpha}\}. \end{aligned} \tag{6}$$

By taking the union over all possible values of $L(-b)$ and applying the union bound, we have

$$\mathbb{P}(W(0) > b) \le \sum_{t=1}^{\infty} \mathbb{P}(L(-b) = -t-b-1, \mathcal{E}_t^{\alpha}). \tag{7}$$

We fix a finite time t^* as

$$t^* \triangleq \max\{T_1, \left\lceil \frac{I_0^{\epsilon}}{I_{BX}} \right\rceil\}, \tag{8}$$

where

$$T_1 \triangleq \max\{T_B(\hat{p} - p, \frac{1-\hat{p}}{6(L+2)}), \frac{18(1+\hat{p})}{(2+\hat{p})(1-\hat{p})}\} \tag{9}$$

and

$$I_{BX} \triangleq \min\{\frac{(1-\hat{p})I_X}{9}, I_B(\hat{p} - p, \frac{1-\hat{p}}{6(L+2)})\}. \tag{10}$$

Then, we split the summation in (7) as

$$\mathbb{P}(W(0) > b) \le P_1 + P_2,$$

where

$$P_1 \triangleq \sum_{t=1}^{t^*} \mathbb{P}(L(-b) = -t-b-1, \mathcal{E}_t^{\alpha}),$$

$$P_2 \triangleq \sum_{t=t^*}^{\infty} \mathbb{P}(L(-b) = -t-b-1, \mathcal{E}_t^{\alpha}).$$

We divide the proof into two parts. In Part 1, we show that there exist a constant $C_1 > 0$ and a finite $N_1 > 0$ such that for all $n \ge N_1$, we have

$$P_1 \le (C_1 + e^{g(n)}) t^* e^{-nI_0^{\epsilon}},$$

where $g(n)$ is a function satisfying that $g(n) \in \omega(f(n)\log^2 n)$ and $g(n) \in o(n)$. And in Part 2, we show that there exists a finite $N_2 > 0$ such that for all $n \ge N_2$, we have

$$P_2 \le 4e^{-nI_0^{\epsilon}}.$$

Finally, combining both Parts, we have

$$\mathbb{P}(W(0) > b) \le \left((C_1 + e^{g(n)})t^* + 4\right) e^{-nI_0^{\epsilon}},$$

for all $n \ge N \triangleq \max\{N_1, N_2\}$. By letting ϵ tend to 0, and taking logarithm and limit as n goes to infinity, we obtain $\liminf_{n\to\infty} \frac{-1}{n} \log \mathbb{P}\left(W(0) > b\right) \ge I_0$, and thus the desired results. For a detailed proof, please refer to our online technical report [24].

REFERENCES

[1] L. Ying, R. Srikant, A. Eryilmaz, and G. Dullerud, "A large deviations analysis of scheduling in wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5088–5098, 2006.

[2] A. Stolyar, "Large deviations of queues sharing a randomly time-varying server," *Queueing Systems*, vol. 59, no. 1, pp. 1–35, 2008.

[3] S. Shakkottai, "Effective capacity and QoS for wireless scheduling," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 749–761, 2008.

[4] V. Venkataramanan and X. Lin, "On wireless scheduling algorithms for minimizing the queue-overflow probability," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 788–801, 2010.

[5] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Scheduling in multi-channel wireless networks: Rate function optimality in the small-buffer regime," in *ACM Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems (SIGMETRICS)*, 2009, pp. 121–132.

[6] C. Courcoubetis and R. Weber, "Buffer overflow asymptotics for a buffer handling many traffic sources," *Journal of Applied Probability*, pp. 886–903, 1996.

[7] S. Shakkottai and R. Srikant, "Many-sources delay asymptotics with applications to priority queues," *Queueing Systems*, vol. 39, no. 2, pp. 183–200, 2001.

[8] S. Kittipiyakul and T. Javidi, "Delay-optimal server allocation in multiqueue multiserver systems with time-varying connectivities," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2319–2333, 2009.

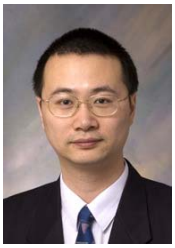
[9] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Low-complexity scheduling algorithms for multi-channel downlink wireless networks," in *The IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2010, pp. 1–9.

[10] ——, "Scheduling for small delay in multi-rate multi-channel wireless networks," in *The IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2011, pp. 1251–1259.
[11] S. Bodas and T. Javidi, "Scheduling for multi-channel wireless networks: Small delay with polynomial complexity," in *2011 International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. IEEE, 2011, pp. 78–85.
[12] M. Sharma and X. Lin, "OFDM downlink scheduling for delay-optimality: Many-channel many-source asymptotics with general arrival processes," in *The IEEE Information Theory and Applications Workshop (ITA)*, 2011.
[13] ——, "OFDM downlink scheduling for delay-optimality: Many-channel many-source asymptotics with general arrival processes," Purdue University, Tech. Rep., 2011. [Online]. Available: https://engineering.purdue.edu/%7elinx/papers.html
[14] B. Ji, C. Joo, and N. B. Shroff, "Delay-Based Back-Pressure Scheduling in Multihop Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1539–1552, 2013.
[15] B. Ji, G. R. Gupta, X. Lin, and N. B. Shroff, "Low-complexity scheduling policies for achieving throughput and asymptotic delay optimality in multi-channel wireless networks," *IEEE/ACM Transactions on Networking*, 2013, accepted for publication.
[16] A. Mekkittikul and N. McKeown, "A starvation-free algorithm for achieving 100% throughput in an input-queued switch," in *Proc. of the IEEE International Conference on Communication Networks (ICCCN)*, 1996.
[17] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications magazine*, vol. 39, no. 2, pp. 150–154, 2001.
[18] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Translations of the American Mathematical Society-Series 2*, vol. 207, pp. 185–202, 2002.
[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 191–217, 2004.
[20] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, 2005.
[21] B. Sadiq and G. de Veciana, "Throughput optimality of delay-driven MaxWeight scheduler for a wireless system with flow dynamics," in *Proceedings of the 47th Annual Conference on Communication, Control and Computing (Allerton)*, 2009.
[22] M. Neely, "Delay-based network utility maximization," in *The 29th IEEE International Conference on Computer Communications (INFOCOM)*, 2010.
[23] M. Bramson, "Stability of queueing networks," *Probability Surveys*, vol. 5, no. 1, pp. 169–345, 2008.
[24] B. Ji, G. R. Gagan, M. Sharma, X. Lin, and N. B. Shroff, "Achieving Optimal Throughput and Near-Optimal Asymptotic Delay Performance in Multi-Channel Wireless Network with Low Complexity: A Practical Greedy Scheduling Policy," *Arxiv preprint arXiv:1212.1638*, November 2013. [Online]. Available: http://arxiv.org/abs/1212.1638
[25] J. Dai, "On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models," *The Annals of Applied Probability*, pp. 49–77, 1995.

Bo Ji (S'11-M'12) received his B.E. and M.E. degrees in Information Science and Electronic Engineering from Zhejiang University, China in 2004 and 2006, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from The Ohio State University. He is currently a Senior Member of Technical Staff at AT&T Labs. Dr. Ji's research interests are in the modeling, analysis, control, and optimization of complex network systems, such as communication networks, data centers, smart grid networks, and cyber-physical networks.

Gagan R. Gupta received the Bachelor of Technology degree in Computer Science and Engineering from the Indian Institute of Technology, Delhi, India in 2005. He received the M.S. degree in Computer Science from University of Wisconsin, Madison in 2006. He received his Ph.D. degree from Purdue University, West Lafayette, Indiana in 2009 for his dissertation titled "delay efficient control policies for wireless networks." His research interests are in performance modeling and optimization of communication networks and parallel computing. He is currently working in the telecommunications industry.

Manu Sharma received his B.Tech. degree in Electrical Engineering from the Indian Institute of Technology Bombay in 2009 and M.S. degree from Purdue University in 2011. He is currently working as Systems Research Engineer at Qualcomm.



Xiaojun Lin (S'02-M'05-SM'12) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, Indiana, in 2000 and 2005, respectively. He is currently an Associate Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin's research interests are in the analysis, control and optimization of wireless and wireline communication networks. He received the IEEE INFOCOM 2008 best paper award and 2005 best paper of the year award from *Journal of Communications and Networks*. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005. He received the NSF CAREER award in 2007. He was the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, the TPC co-chair for ACM MobiHoc 2009, and the Mini-Conference co-chair for IEEE INFOCOM 2012. He is currently serving as Associated Editor for IEEE/ACM Transactions on Networking, an Area Editor for (Elsevier) Computer Networks journal, and has served as a Guest Editor for (Elsevier) Ad Hoc Networks journal.

Ness B. Shroff (S'91-M'93-SM'01-F'07) received his Ph.D. degree in Electrical Engineering from Columbia University in 1994. He joined Purdue university immediately thereafter as an Assistant Professor in the school of Electrical and Computer Engineering. At Purdue, he became Full Professor of ECE in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined The Ohio State University, where he holds the Ohio Eminent Scholar endowed chair in Networking and Communications, in the departments of ECE and CSE. His research interests span the areas of communication, social, and cyberphysical networks. He is especially interested in fundamental problems in the design, control, performance, pricing, and security of these networks. Dr. Shroff is a Fellow of the IEEE, serves on a number of editorial boards of journals, and has received numerous best paper awards for his research.