Low-Complexity Scheduling Policies for Achieving Throughput and Asymptotic Delay Optimality in Multichannel Wireless Networks

Bo Ji, *Member, IEEE*, Gagan R. Gupta, Xiaojun Lin, *Senior Member, IEEE*, and Ness B. Shroff, *Fellow, IEEE*

***Abstract*—In this paper, we study the scheduling problem for downlink transmission in a multichannel (e.g., OFDM-based) wireless network. We focus on a single cell, with the aim of developing a unifying framework for designing low-complexity scheduling policies that can provide optimal performance in terms of both throughput and delay. We develop new easy-to-verify sufficient conditions for rate-function delay optimality (in the many-channel many-user *asymptotic* regime) and throughput optimality (in general *nonasymptotic* setting), respectively. The sufficient conditions allow us to prove rate-function delay optimality for a class of Oldest Packets First (OPF) policies and throughput optimality for a large class of Maximum Weight in the Fluid limit (MWF) policies, respectively. By exploiting the special features of our carefully chosen sufficient conditions and intelligently combining policies from the classes of OPF and MWF policies, we design hybrid policies that are both rate-function delay-optimal and throughput-optimal with a complexity of $O(n^{2.5}\log n)$, where n is the number of channels or users. Our sufficient condition is also used to show that a previously proposed policy called Delay Weighted Matching (DWM) is rate-function delay-optimal. However, DWM incurs a high complexity of $O(n^5)$. Thus, our approach yields significantly lower complexity than the only previously designed delay and throughput-optimal scheduling policy. We also conduct numerical experiments to validate our theoretical results.**

***Index Terms*—Delay optimality, large-deviations theory, low-complexity, multichannel, OFDM, quality of service, scheduling, throughput optimality, wireless networks.**

I. Introduction

Designing high-performance scheduling algorithms has been a vital and challenging problem in wireless networks. Among the many dimensions of network performance, the most critical ones are perhaps throughput, delay, and complexity. However, it is in general extremely difficult, if not impossible, to develop scheduling policies that attain the optimal performance in terms of both throughput and delay, without the cost of high complexity [1].

In this paper, we focus on the setting of a single-hop multiuser multichannel system. A practically important example of such a multichannel system is the downlink of a single cell in 4G OFDM-based celluar networks (e.g., LTE and WiMAX). Such a system typically has a large bandwidth that can be divided into multiple orthogonal subbands (or channels), which need to be allocated to a large number of users by a scheduling algorithm. The main question that we will attempt to answer in this paper is the following: *How do we design efficient scheduling algorithms that simultaneously provide high throughput, small delay, and low complexity?*

We consider a multichannel system that has n channels and a proportionally large number of users. This setting is referred to as the many-channel many-user asymptotic regime when n goes to infinity. The connectivity between each user and each channel is assumed to be time-varying, due to channel fading. We assume that the base station (BS) maintains separate first-in–first-out (FIFO) queues that buffer the packets destined to each user. The *delay* metric that we will focus on in this paper is the *asymptotic decay-rate* (also called the *rate-function* in the large-deviations theory) of the probability that the largest packet waiting time in the system exceeds a fixed threshold, as both the number of channels and the number of users go to infinity. [Refer to (2) for the precise definition.]

Next, we overview some key related works. In [2], the authors considered a single-server model with time-varying channels and showed that the longest-connected-queue (LCQ) algorithm minimizes the average delay for the special case of symmetric (*i.i.d.* Bernoulli) arrival and channel. Later, the results were generalized for a multiserver model in [3]. The authors of [4] further generalized the multiserver model by considering more general permutation-invariant arrivals (that are not restricted to Bernoulli only) and multirate channel model. Hence, the problem of minimizing a general cost function of queue lengths (includes minimizing the expected delay) studied in [4] becomes harder. There, for special cases of ON–OFF channel model with two users or allowing for fractional server allocation, an optimal scheduling algorithm was derived. Using the insights obtained from the analytical results in [4] for ON–OFF channel model, in [5] the same authors developed heuristic policies and showed through simulations that their proposed heuristic policies perform well under a general channel model. Note that in contrast to this paper, the

Manuscript received November 20, 2012; revised August 05, 2013; accepted October 15, 2013; approved by IEEE/ACM Transactions on Networking Editor U. Ayesta. This work was supported in part by the ARO MURI under Grant W911NF-08-1-0238 and the NSF under Grants CNS-1065136, CNS-1012700, and CNS-0643145.

B. Ji is with AT&T Labs, San Ramon, CA 94582 USA (e-mail: ji.33@osu.edu).

G. R. Gupta and X. Lin are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: gagan.gupta@iitdalumni.com; linx@ecn.purdue.edu).

N. B. Shroff is with the Departments of Electrical and Computer Engineering and Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: shroff.11@osu.edu).

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/TNET.2013.2291793

above studies directly minimize queue length or delay in a *nonasymptotic* regime, which is an extremely difficult problem in general.

As we do in this paper, another body of related works [6]–[9] focuses on the many-channel many-user asymptotic regime, where the analysis may become more tractable. Even though the analysis for an asymptotic setting is very different from the nonasymptotic analysis in [4], it is remarkable that some of the insights are consistent. For example, from a delay optimality perspective, the above two bodies of studies both point to the tradeoff between maximizing instantaneous throughput and balancing the queues. Thus, we believe that, collectively, these studies under different settings provide useful insights for designing efficient scheduling solutions in practice.

In [6]–[9], a number of queue-length-based scheduling policies for achieving optimal or positive queue-length-based rate-function[1] were developed. In particular, an optimal scheduling policy that maximizes the queue-length-based rate-function has been derived with complexity $O(n^3)$ [9]. However, these works have two key limitations. First, the schedulers' performance are proven under the assumption that the arrival process is *i.i.d. not only across users, but also in time*, which does not model the temporal correlation present in most real network traffic. More importantly, it is well known that good queue-length performance does not necessarily translate to good delay performance [10]–[12]. A recently developed scheduling policy called Delay Weighted Matching (DWM) [10], [11], which makes scheduling decisions by maximizing the sum of the delays of the scheduled packets in each time-slot, focuses directly on the delay performance as we do in this paper. It has been shown that DWM is rate-function delay-optimal in some cases. However, DWM has the following two key drawbacks: 1) it is unclear whether DWM is rate-function delay-optimal in general; and 2) DWM yields a very high complexity of $O(n^5)$ and is thus not amenable for practical implementations.

Hence, the state of the art does not satisfactorily answer our main question of how to design scheduling policies with a low complexity, while guaranteeing *provable optimality* for both throughput and delay. In this paper, we address this challenge and provide the following key intellectual contributions.

First, we characterize *easy-to-verify* sufficient conditions for rate-function delay optimality in the many-channel many-user asymptotic regime and for throughput optimality in general nonasymptotic settings. The sufficient conditions allow us to prove rate-function delay optimality for a class of *Oldest Packets First* (OPF) policies and throughput optimality for a large class of *Maximum Weight in the Fluid limit* (MWF) policies. Moreover, the sufficient conditions can be used to show that a slightly modified version of the DWM policy is both rate-function delay-optimal and throughput-optimal.

Second, we develop an $O(n^{2.5}\log n)$-complexity scheduling policy called DWM-n. The DWM-n policy shares the high-level similarity with the DWM policy [10], [11], but makes scheduling decisions in each time-slot by maximizing the sum of the delays of the scheduled packets over only the n oldest packets in the system, rather than over all the packets as in the DWM policy. We show that DWM-n is an OPF policy and is thus rate-function delay-optimal. However, DWM-n is *not* throughput-optimal in general and may perform poorly when n is not large.

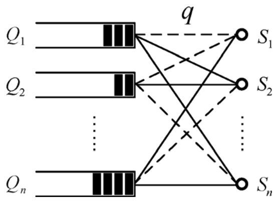

Fig. 1. System model. The connectivity between each pair of queue Q_i and server S_j is "ON" (denoted by a solid line) with probability q, and "OFF" (denoted by a dashed line) otherwise.

Third, by exploiting the special features of our carefully chosen sufficient conditions and intelligently combining policies from the classes of OPF and MWF policies, we develop a class of two-stage hybrid policies that are both throughput-optimal and rate-function delay-optimal. In particular, we can adopt the DWM-n policy in stage 1 and the Delay-based MaxWeight Scheduling (D-MWS) policy in stage 2, respectively, so as to *design an optimal hybrid policy with a low complexity of* $O(n^{2.5}\log n)$.

Finally, we conduct numerical experiments to validate our theoretical results in different scenarios.

II. System Model

We consider a multichannel system with n orthogonal channels and n users, which can be modeled as a multiqueue multiserver system with stochastic connectivity, as shown in Fig. 1. *For ease of presentation, the number of users is assumed to be equal to the number of channels. Our analysis for rate-function delay optimality follows similarly if the number of users scales linearly with the number of channels*. Throughout the rest of the paper, we will use the terms "user" and "queue" interchangeably, and use the terms "channel" and "server" interchangeably. We assume that time is slotted. In a time-slot, a server can be allocated to only one queue, but a queue can get service from multiple servers. The connectivity between queues and servers is time-varying, i.e., it can change between "ON" and "OFF" from time to time. We assume that 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 multichannel cellular system with dedicated feedback channels.

The notations used in this paper are as follows. We let Q_i denote the FIFO queue (at the BS) associated with the ith user, and let S_j denote the jth server. We assume infinite buffer for all the queues. 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 be the mean arrival rate of queue Q_i, and let $\lambda \triangleq [\lambda_1, \lambda_2, \ldots, \lambda_n]$ denote the arrival rate vector. We assume that packet arrivals occur at the beginning of each time-slot, and packet departures occur at the end of each time-slot. We let $Q_i(t)$ denote the length of queue

[1] The queue-length-based rate-function is defined as the asymptotic decay-rate of the probability that the largest queue length in the system exceeds a fixed threshold.

Q_i at the beginning of time-slot t immediately after packet arrivals. Also, let $Z_{i,l}(t)$ denote the delay (i.e., waiting time) of the lth packet of queue Q_i at the beginning of time-slot t, *which is measured since 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 still present in the system will have their delays increased by one due to the elapsed time. We then let $W_i(t) = Z_{i,1}(t)$ denote the head-of-line (HOL) packet delay of queue Q_i at the beginning of time-slot t. Furthermore, we use $C_{i,j}(t)$ to denote the capacity of the link between queue Q_i and server S_j in time-slot t, i.e., the maximum number of packets that can be served by server S_j from queue Q_i in time-slot t. Finally, we let $\mathbb{1}_{\{\cdot\}}$ denote the indicator function, and let $\mathbb{Z}^+$ denote the set of positive integers.

We now state the assumptions on the arrival processes. The throughput analysis is carried out under very general conditions (Assumption 1) similar to that of [13].

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 showing throughput optimality, as discussed in [13]).

Assumptions 2 and 3 will be used for rate-function delay analysis.

Assumption 2: There exists a finite L such that $A_i(t) \leq L$ for any i and t, i.e., arrivals are bounded. Furthermore, we assume $\mathbb{P}(A(s, s+t-1) = Lnt) > 0$ for any s, t, and n.

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 exists $T_B(\epsilon, \delta) > 0$, $N_B(\epsilon, \delta) > 0$, and a positive function $I_B(\epsilon, \delta)$ independent of n and t such that

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

for all $t \geq T_B(\epsilon, \delta)$ and $n \geq N_B(\epsilon, \delta)$.

Assumptions 2 and 3 are relatively mild. The first part of Assumptions 2 and 3 have also been used in the previous work [10], [11] for rate-function delay analysis. In Assumption 2, the first part requires that the arrivals in each time-slot have bounded support; the second part guarantees that there is a positive probability that all users have the maximum number of arrivals in any time-interval with any length. Assumption 3 allows the arrivals for each user to be correlated over time (e.g., arrivals driven by a two-state Markov chain), which is more general than the arrival processes (*i.i.d.* in time) considered in [6]–[9].

We then describe our channel model as follows.

Assumption 4: In any time-slot t, $C_{i,j}(t)$ is modeled as a Bernoulli random variable with a parameter $q \in (0, 1)$, i.e.,

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

All the random variables $C_{i,j}(t)$ are assumed to be mutually independent across all the variables i, j, and t.

We assume unit channel capacity as above. Under this assumption, we will also let $C_{i,j}(t)$ denote the connectivity between queue Q_i and server S_j in time-slot t, without causing confusions. As in the previous works [6]–[11], in this paper we assume *i.i.d.* channels for the analytical results only. Moreover, we will show through simulations that our proposed low-complexity solution also performs well in more general scenarios, e.g., when the channel condition follows a two-state Markov chain that allows correlation over time. Furthermore, we will briefly discuss how to generalize our solution to more general scenarios toward the end of this paper.

Next, we define the *optimal throughput region* (or *stability region*) of the system for any fixed integer $n > 0$. As in [13], 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. 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. Furthermore, the *optimal throughput region* is defined as the union of the throughput regions of all possible scheduling policies. We let Λ^* denote the optimal throughput region. A scheduling policy is *throughput-optimal* if it can stabilize any arrival rate vector λ strictly inside Λ^*. For more discussions on the characterization of Λ^*, please refer to our online technical report [14].

For delay analysis, we consider the many-channel many-user asymptotic regime. Let $W(t)$ denote the largest HOL delay over all the queues (i.e., the largest or worst packet waiting time in the system) at the beginning of time-slot t, i.e., $W(t) \triangleq \max_{1\leq i\leq n} W_i(t)$. Assuming that the system is stationary and ergodic, we define the *rate-function* for integer threshold $b \geq 0$ as

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

We can then estimate $\mathbb{P}(W(0) > b) \approx \exp(-nI(b))$ when n is large, and the estimation accuracy tends to be higher as n increases. Clearly, for large n, a larger value of the rate-function leads to better delay performance, i.e., a smaller probability that the largest HOL delay exceeds a certain threshold. A scheduling policy is *rate-function delay-optimal* if for any fixed integer threshold $b \geq 0$, it achieves the maximum rate-function over all possible scheduling policies.

Note that *the rate-function optimality is studied in the asymptotic regime, i.e., when n goes to infinity. Although the convergence of the rate-function is typically fast, the throughput performance may be poor for small to moderate values of n.* As a matter of fact, a rate-function delay-optimal policy may not even be throughput-optimal for a fixed n (e.g., the DWM-n policy that we will propose in Section IV). To that end, we are interested in designing scheduling policies that maximize both the throughput (for any fixed n) and the rate-function (in the many-channel many-user asymptotic regime).

III. Upper Bound on the Rate-Function

In this section, we derive an upper bound on the rate-function that can be achieved by any scheduling algorithm. Then, in

Section IV, we will provide a sufficient condition for achieving this upper bound and thus achieving the optimal rate-function.

As in [10] and [11], for any integer $t > 0$ and any real number $x \geq 0$, we define the quantity

$$I_A(t,x) \triangleq \sup_{\theta > 0} \left[\theta(t+x) - \lambda_{A_i(-t+1,0)}(\theta)\right]$$

where $\lambda_{A_i(-t+1,0)}(\theta) = \log \mathbf{E}[e^{\theta A_i(-t+1,0)}]$ is the cumulant-generating function of $A_i(-t+1,0)$ and $A_i(-t+1,0) = \sum_{\tau=-t+1}^{0} A_i(\tau)$. From Cramer's Theorem, this quantity, $I_A(t,x)$, is equal to the asymptotic decay-rate of the probability that in any interval of t time-slots, the total number of packet arrivals to the system is no smaller than $n(t+x)$, as n tends to infinity, i.e.,

$$\lim_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(A(-t+1,0) \geq n(t+x)) = I_A(t,x). \quad (3)$$

Define the following for the case of $L > 1$. For any integer $x \geq 0$, we define t_x as

$$t_x \triangleq \frac{x}{L-1}.$$

Then, we define $\Psi_b \triangleq \{c \in \{0,1,\ldots,b\} | t_{b-c} \in \mathbb{Z}^+\}$. It will later become clear why the values of c in the set Ψ_b are important and need to be considered separately. Let $I_X \triangleq \log \frac{1}{1-q}$. Then, for any integer $b \geq 0$, we define the quantity

$$\begin{aligned} I_0(b) \triangleq \min\Bigg\{&(b+1)I_X, \\ &\min_{0\leq c\leq b}\left\{\inf_{t>t_{b-c}} I_A(t,b-c)+cI_X\right\}, \\ &\min_{c\in\Psi_b}\{I_A(t_{b-c},b-c)+(c+1)I_X\}\Bigg\}. \quad (4)\end{aligned}$$

Furthermore, for any given integer $L \geq 1$, we define

$$I_0^*(b) \triangleq \begin{cases} (b+1)I_X, & \text{if } L=1 \\ I_0(b), & \text{if } L>1. \end{cases}$$

In the following theorem, we show that for any given integer threshold $b \geq 0$, $I_0^*(b)$ is an upper bound of the rate-function that can be achieved by any scheduling policy.

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

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq I_0^*(b)$$

for any given integer threshold $b \geq 0$.

We prove Theorem 1 by considering three types of events that lead to the delay-violation event $\{W(0) > b\}$ and computing their probabilities. We provide the proof in Appendix A.

Note that in [10], the authors derived another upper bound $\min\{(b+1)I_X, \min_{0\leq c\leq b}\{I_A^+(b-c)+cI_X\}\}$, where $I_A^+(x) \triangleq \inf_{t>0} I_A^+(t,x)$ and $I_A^+(t,x) \triangleq \lim_{y\to x^+} I_A(t,y)$. We would like to remark that their upper bound was derived by considering two types of events that lead to the delay-violation event, which yet accounts for only a proper subset of the events that we consider in Appendix A. Hence, their upper bound could be larger than $I_0^*(b)$ in some cases.

IV. Sufficient Conditions

In [10] and [11], the authors proposed the DWM policy and studied its rate-function delay optimality[2] (without the second part of Assumption 2) in some cases. Specifically, in [10] and [11], the authors proved that DWM attains a rate-function that is no smaller than $\min\{(b+1)I_X, \min_{0\leq c\leq b}\{I_A(b-c)+cI_X\}\}$, where $I_A(x) \triangleq \inf_{t>0} I_A(t,x)$. This is proved by showing that the FBS policy (with a properly chosen operating parameter h) can attain this rate-function and DWM dominates FBS for all values of h in a sample-path sense. As pointed out in [10, Sec. V.D], there may be a gap between the rate-function attained by DWM and the upper bound derived in [10], depending on the value of b and the arrival process. More specifically, it can be shown that for given $b \geq 0$, if $I_A(b-c) = I_A^+(b-c)$ for all values of $c \in \{0,\ldots,b\}$ for the given arrival process, then both FBS and DWM are rate-function delay-optimal.

However, it is unclear whether the DWM policy is rate-function delay-optimal in general. Moreover, its high complexity $O(n^5)$ renders it impractical. Hence, *the grand challenge is to find low-complexity scheduling policies that are both throughput-optimal and rate-function delay-optimal.* To that end, in this section, we first characterize easy-to-verify sufficient conditions for rate-function delay optimality in the many-channel many-user asymptotic regime and for throughput optimality in nonasymptotic settings. We then develop two classes of policies, called the OPF policies and the MWF polices, that satisfy the sufficient condition for rate-function delay optimality and throughput optimality, respectively.

As discussed in the Introduction, our ultimate goal is to *develop low-complexity hybrid policies that are both rate-function delay-optimal and throughput-optimal.* However, it is unclear that just because one policy is rate-function delay-optimal and another one is throughput-optimal, their combinations will necessarily yield the right hybrid policy that is optimal in terms of both throughput and delay. As we will discuss further in the beginning of Section V, our carefully chosen sufficient conditions possess some special features that allow us to construct low-complexity hybrid policies that are both rate-function delay-optimal and throughput-optimal.

A. Rate-Function Delay Optimality

We start by presenting the main result of this section in the following theorem, which provides a sufficient condition for scheduling policies to be rate-function delay-optimal.

Theorem 2: Under Assumptions 2 and 3, a scheduling policy $\mathbf{P}$ is rate-function delay-optimal if in any time-slot, policy $\mathbf{P}$ can serve the k oldest packets in that time-slot for the largest possible value of $k \in \{1,2,\ldots,n\}$.

To prove Theorem 2, we will exploit a dominance property (Lemma 3) of the policies that satisfy the above sufficient condition. Due to space constraint, we have provided the full proof with all the details in our online technical report [14]. However, in this paper we do provide an outline of the proof in Appendix B and give the intuition behind it as follows. First, it is easy to see that the first-come–first-serve (FCFS)

[2] Although the delay metric considered in [10] and [11] is slightly different from ours, both metrics are closely related. Moreover, the rate-function delay analysis for DWM in [10] and [11] is also applicable for our defined rate-function as in (2).

policy, which serves the *oldest packets first*, is (sample-path) delay-optimal in a single-queue single-server system. Also, it is not hard to see that for a multiqueue multiserver system with *full connectivity*, where all pairs of queue and server are connected, a policy that chooses to serve the oldest packets (over the whole system) first is delay-optimal. These motivate us to ask a natural and interesting question: *If a policy chooses to serve the oldest packets first in a multiqueue multiserver system with time-varying and partial connectivity (as we consider in this paper), does it achieve rate-function delay optimality?* Note that in such a system, at most n packets can be served in each time-slot. Hence, if in each time-slot a policy can serve all the n oldest packets in the system (as in the case with full connectivity), this policy should yield optimal delay performance. However, due to the random connectivity between queues and servers, no policy may be able to do so. Hence, we propose a class of policies that choose to serve the k oldest packets for the largest possible value of k. In other words, for any $k \in \{1, 2, \dots, n\}$, if the k oldest packets can be served by some scheduling policy, then our proposed policies will serve these k packets too.

A similar, but less thorough, analysis was also carried out in [10] and [11]. There, the authors proposed the *Frame-Based Scheduling* (FBS) policy, which aims to serve the oldest packets in each time-slot and can be viewed as an approximation of FCFS policy. The FBS policy serves packets in units of frames. With a given positive integer h as the operating parameter, each frame is constructed such that: 1) the difference of the arrival times of any two packets within a frame must be no greater than h; and 2) the total number of packets in each frame is no greater than $n_0 = n - Lh$. In each time-slot, the packets arrived at the beginning of this time-slot are filled into the last frame until any of the above two conditions are violated, in which case a new frame will be opened. In any time-slot, the FBS policy serves the HOL frame that contains the oldest (up to n_0) packets with high probability for large n. As discussed at the beginning of this section, it has been shown that the FBS policy with a properly chosen operating parameter h is rate-function delay-optimal in some cases.

However, FBS may *not* be rate-function delay-optimal in some other cases. Specifically, consider *i.i.d.* Bernoulli arrivals with $L = 1$. As pointed out in [10], the rate-function attained by the FBS policy is not optimal in this scenario. We provide the intuition as follows. Suppose there are a total of nt packet arrivals to the system in an interval of t time-slots. It is easy to see that FBS needs at least $t + 1$ time-slots to completely serve these packets since at most $n - Lh$ packets can be served by FBS in one time-slot. This could lead to a suboptimal rate-function. To see this, consider the *perfect-matching* policy defined as follows. Let $\mathcal{Q}$ and $\mathcal{S}$ denote the set of queues and set of servers, respectively. In a time-slot τ, let $\mathcal{C} \triangleq \{C_{i,j}(\tau) : C_{i,j}(\tau) = 1\}$ denote the set of edges between $\mathcal{Q}$ and $\mathcal{S}$. Clearly, $G[\mathcal{Q} \cup \mathcal{S}, \mathcal{C}]$ forms a bipartite graph. If a perfect matching can be found in the bipartite graph $G[\mathcal{Q} \cup \mathcal{S}, \mathcal{C}]$, then the servers are allocated to serve the oldest packets in the respective queues as determined by the perfect matching. Otherwise, none of the servers will be allocated to the queues. It has been shown in [6] that in each time-slot, a perfect matching can be found with high probability for large n. Hence, in the case described above, the perfect-matching policy needs only t time-slots to drain all these nt packets with high probability for large n, while FBS is suboptimal.

On the other hand, the perfect-matching policy does not perform well in many other cases due to the fact that it cannot serve more than one packet from each queue in a time-slot. For example, consider the case where there are L packets existing in Q_1 and the other queues are all empty. FBS can drain these packets within one time-slot with high probability, yet the perfect-matching policy needs at least L time-slots.

The above discussions suggest that if we can find a policy that dominates both the FBS policy and the perfect-matching policy, there is hope that this policy may be able to achieve the optimal rate-function in general. We will show in Lemma 3 that a policy that satisfies the sufficient condition in Theorem 2 indeed dominates both the FBS policy and the perfect-matching policy in a sample-path sense.

In order to state the dominance property of Lemma 3, we consider the following versions of the FBS policy and the perfect-matching policy. Suppose that packet p is the x_pth arrival to the queue $Q_{q(p)}$ in time-slot t_p. Then, we define the weight of the packet p in time-slot t as $\hat{w}(p) = t - t_p + \frac{L+1-x_p}{L+1} + \frac{n+1-q(p)}{(L+1)(n+1)}$. For two packets p_1 and p_2, we say p_1 is older than p_2 if $\hat{w}(p_1) > \hat{w}(p_2)$. The above way of defining the weight ensures that among the packets that arrive at the same time, the priority is given to the packet that has an earlier order of arrival in each queue; and furthermore, among the packets (in different queues) with the same order of arrival, the priority is given to the packet that arrives to the queue with a smaller index. For the FBS policy, we assume that the packets with a larger weight are filled to the frame with a higher priority when there are multiple packets arriving at the same time. Meanwhile for the perfect-matching policy, we require that in time-slot t, the perfect-matching policy only serves packets with the largest value of $t - t_p + \frac{L+1-x_p}{L+1}$. Under this version of the perfect-matching policy, it is possible that a queue may not have any of its packets served even if a perfect-matching is found and a server is allocated to the queue. It should be noted that the above versions of the FBS policy and the perfect-matching policy are used for analysis only. Next, we present the dominance property in the following lemma.

Lemma 3: Consider the versions of the FBS policy and the perfect-matching policy described above. Suppose that policy $\mathbf{P}$ satisfies the sufficient condition in Theorem 2. Then, for any given sample path, by the end of any time-slot t, policy $\mathbf{P}$ has served every packet that the FBS policy or the perfect-matching policy has served.

We prove Lemma 3 by contradiction and provide the proof in Appendix C. Furthermore, by using this dominance property, and following a similar argument as in the rate-function delay analysis for FBS ([10, Theorem 2]), we prove Theorem 2. Specifically, we consider all the sample paths that lead to the delay-violation event. There are different ways that the delay-violation event can occur, each of which 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"). An outline of the proof for Theorem 2 is provided in Appendix B.

Next, we define a class of OPF policies as follows.

Definition 1: A scheduling policy $\mathbf{P}$ is said to be in the class of OPF policies if policy $\mathbf{P}$ satisfies the sufficient condition in Theorem 2.

Clearly, the class of OPF policies is all rate-function delay-optimal. We would like to emphasize that the sufficient condition in Theorem 2 is very easy to verify and can be readily used to design other rate-function delay-optimal policies. Specifically, Theorem 2 enables us to identify a new rate-function delay-optimal policy, called the $\mathbf{DWM}-n$ policy, which substantially reduces the complexity to $O(n^{2.5}\log n)$. This in turn allows us to design *low-complexity* hybrid scheduling policies that are both *throughput-optimal* and *rate-function delay-optimal* (in Section V).

Now, we review the DWM policy proposed in [10] and [11]. DWM operates in the following way. In each time-slot t, define the weight of the lth packet of Q_i as $Z_{i,l}(t)$, i.e., the delay of this packet at the beginning of time-slot t, which is measured since the time when this packet arrived to queue Q_i until time-slot t. Then, construct a bipartite graph $G[X \cup Y, E]$ such that the vertices in X correspond to the n oldest packets from each of the n queues and Y is the set of all servers. Thus, $|X| = n^2$ and $|Y| = n$. Let $X_i \subseteq X$ be the set of packets from queue Q_i. If queue Q_i is connected to server S_j, then for each packet $x \in X_i$, there is an edge between x and S_j in graph G, and the weight of this edge is set to the weight of packet x. The schedule is then determined by a maximum-weight matching over G. Clearly, DWM maximizes the sum of the delays of the packets scheduled.

It has been shown in [10] and [11] that the DWM policy is rate-function delay-optimal in some cases. However, it is unclear whether it is delay-optimal in general. *We would like to highlight that our proposed sufficient condition in Theorem 2 allows us to show that a slightly modified version of the DWM policy is rate-function delay-optimal in general (under an additional mild assumption—the second part of Assumption 2).* Specifically, in the modified version of the DWM policy, we assign the weight of a packet p as $\hat{w}(p)$ instead of its delay only. Then, by simply duplicating the proof of [10, Lemma 7], we can show that the modified version of the DWM policy is an OPF policy and is thus rate-function delay-optimal.

However, the DWM policy still suffers from a high complexity, which renders it impractical. Specifically, DWM has a complexity of $O(n^5)$ since the complexity of finding a maximum-weight matching [15] over a bipartite graph $G[V, E]$ is $O(|V||E|+|V|^2\log|V|)$ in general, and the bipartite graph constructed by DWM has $|V| = O(n^2)$ and $|E| = O(n^3)$.

To overcome the high-complexity issue, we develop a simpler policy that is also in the class of the OPF policies (and is thus rate-function delay-optimal), but has a much lower complexity of $O(n^{2.5}\log n)$. The new policy is called the $\mathbf{DWM}-n$ policy due to the high-level similarity with DWM. However, it exhibits critical differences when picking packets to construct the bipartite graph $G[X \cup Y, E]$ and finding the maximum-weight matching over G. The differences are as follows.

1) In each time-slot, instead of considering the n oldest packets from each queue (and thus n^2 packets in total) as in DWM, DWM-n considers only the n oldest packets in the whole system. Hence, the bipartite graph constructed by DWM-n has $|X| = n$ and $|Y| = n$.
2) The rest of the operations of DWM-n are similar to that of DWM, i.e., the schedule is determined by a maximum-weight matching over G, except that DWM-n finds a maximum-weight matching based on the *vertex* weights. Such a maximum-weight matching is also called Maximum Vertex-weighted Matching (MVM) [16], [17]. Specifically, the weight of each vertex $p \in X$ is set to $\hat{w}(p)$ (i.e., the weight of the corresponding packet p), and the weight of each vertex in the set Y is set to 0.

In the following proposition, we show that the DWM-n policy is rate-function delay-optimal and has a low complexity.

Proposition 4: The DWM-n policy is an OPF policy and is thus rate-function delay-optimal under Assumptions 2 and 3. Furthermore, the DWM-n policy has a low complexity of $O(n^{2.5}\log n)$.

We provide the proof in Appendix D. The fact that the DWM-n policy is an OPF policy follows from a property of MVM [16] that if there exists a matching that matches all of the k heaviest vertices, then any MVM matches all of the k heaviest vertices as well. The low complexity of DWM-n follows immediately from the fact that DWM-n reduces the number of packets under consideration (n packets in total), and that an MVM in an $n \times n$ bipartite graph can be found in $O(n^{2.5}\log n)$ time [16]. Note that even if the DWM policy adopts MVM when determining the schedule, its complexity can only be reduced to $O(n^4\log n)$.

Although the DWM-n policy achieves rate-function delay optimality with a low complexity, it may not be throughput-optimal in general. This is because the DWM-n policy considers only the n oldest packets in the system. It is likely that certain servers may not be connected to any of the queues that contain these n packets, which results in the server being idle and is thus a waste of service. Hence, DWM-n is a lazy policy. In fact, we can construct a simple counterexample to show that the DWM-n policy is, in general, not throughput-optimal as stated in Proposition 5.

Proposition 5: The DWM-n policy is not throughput-optimal in general.

We prove Proposition 5 by constructing a special arrival pattern that forces certain servers to be idle, even when they can serve some of the queues. We provide the proof in Appendix E. Proposition 5 suggests that a rate-function delay-optimal policy may not have good throughput performance (for a fixed n). This may appear counterintuitive at the first glance. However, it should be noted that the rate-function delay optimality is studied in the asymptotic regime, i.e., when n goes to infinity. Although the convergence of the rate-function is typically fast, the throughput performance may be poor for small to moderate values of n. Our simulation results (Fig. 3 in Section VI) will provide further evidence of this.

B. Throughput Optimality

In this section, we present a sufficient condition for throughput optimality in very general nonasymptotic settings.

Recall that $Q_i(t)$ denotes the length of queue Q_i at the beginning of time-slot t immediately after packet arrivals, $Z_{i,l}(t)$ denotes the delay of the lth packet of Q_i at the beginning of time-slot t, $W_i(t) = Z_{i,1}(t)$ denotes the HOL packet delay of Q_i at the beginning of time-slot t, and $C_{i,j}(t)$ denotes the connectivity between Q_i and S_j in time-slot t. Let $\mathcal{S}_j(t)$ denote the set of queues being connected to server S_j in time-slot t, i.e., $\mathcal{S}_j(t) = \{1 \le i \le n | C_{i,j}(t) = 1\}$, and let $\Gamma_j(t)$ denote the subset of queues in $\mathcal{S}_j(t)$ that have the largest weight in time-slot t, i.e., $\Gamma_j(t) \triangleq \{i \in \mathcal{S}_j(t) | W_i(t) = \max_{l \in \mathcal{S}_j(t)} W_l(t)\}$. We now present the main result of this section.

Theorem 6: Let $i(j,t)$ be the index of the queue that is served by server S_j in time-slot t, under a scheduling policy $\mathbf{P}$. Under Assumption 1, policy $\mathbf{P}$ is throughput-optimal if there exists a constant $M > 0$ such that, in any time-slot t and for all $j \in \{1, 2, \dots, n\}$, queue $Q_{i(j,t)}$ satisfies that $W_{i(j,t)}(t) \ge Z_{r,M}(t)$ for all $r \in \Gamma_j(t)$ such that $Q_r(t) \ge M$.

We prove Theorem 6 using fluid limit techniques [13], [18] and standard Lyapunov argument. Due to space constraint, we provide the proof in our online technical report [14]. The condition in Theorem 6 means the following: In each time-slot, *each server chooses to serve a queue with HOL packet delay no less than the delay of the Mth packet in the queue with the largest HOL delay (among the queues connected to the server)*; if this queue (with the largest HOL delay) has less than M packets, then the server may choose to serve any queue.

It is well-known that the MaxWeight Scheduling (MWS) policy [12], [13], [19]–[22] that maximizes the weighted sum of the rates, where the weights are queue lengths or delays, is throughput-optimal in very general settings, including the multichannel system that we consider in this paper. The intuition behind Theorem 6 is that to achieve throughput optimality in our multichannel system, it is sufficient for each server to choose a connected queue with a large enough weight such that this queue has the largest weight in the fluid limit. This relaxes the condition that each server has to find a queue with the largest weight in the original system, and thus significantly expands the set of known throughput-optimal policies.

Next, we define the class of MWF policies as follows.

Definition 2: A policy $\mathbf{P}$ is said to be in the class of MWF policies if policy $\mathbf{P}$ satisfies the sufficient condition in Theorem 6.

Clearly, the class of MWF policies is all throughput-optimal. It is claimed in [10] and [11] that the DWM policy is throughput-optimal, yet the throughput optimality was not explicitly proved there. For completeness, we state the following proposition on throughput optimality of the DWM policy and provide its proof in our online technical report [14].

Proposition 7: The DWM policy is an MWF policy and is thus throughput-optimal under Assumption 1.

Next, we study a simple extension of the delay-based MaxWeight policy [12], [13], [22] that is throughput-optimal in our multichannel system.

D-MWS Policy: In each time-slot t, the scheduler allocates each server S_j to serve queue $Q_{i(j,t)}$ such that $i(j,t) = \min\{i | i \in \Gamma_j(t)\}$. In other words, each server chooses to serve a queue that has the largest HOL delay (among all the queues connected to this server), breaking ties by picking the one with the smallest index if there are multiple such queues.

It is easy to see that D-MWS is an MWF policy and is thus throughput-optimal. Also, it is worth noting that D-MWS has a low complexity of $O(n^2)$ in our multichannel system. However, we can show that D-MWS suffers from poor delay performance. Specifically, we show in the following proposition that under D-MWS, the probability that the largest HOL delay exceeds any fixed threshold, is at least a constant, even if n is large. This results in a zero rate-function.

Proposition 8: Consider *i.i.d.* Bernoulli arrivals, i.e., in each time-slot, and for each user, there is a packet arrival with probability p, and no arrivals otherwise. By allocating servers to queues according to D-MWS, we have that

$$\limsup_{n \to \infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) = 0 \tag{5}$$

for any fixed integer $b \ge 0$.

Due to space constraint, we provide the proof in our online technical report [14] and explain the intuition behind it in the following. Note that under D-MWS, each server chooses to serve a connected queue having the largest weight without accounting for the decisions of the other servers. This way of allocating servers may incur an unbalanced schedule such that in each time-slot, with high probability, only a small fraction of the queues ($O(\log n)$ out of n queues) gets served, while the number of queues having arrivals is much larger $(O(n))$. This then leads to poor delay performance. By an argument similar to that of [7, Theorem 3] [where the authors show that the Queue-length-based MaxWeight Scheduling (Q-MWS) policy results in a zero queue-length rate-function], we can show that under D-MWS, the delay-violation event occurs with at least a constant probability for any fixed threshold even if n is large.

We conclude this section with a summary of the scheduling policies proposed and/or discussed in this section. The FBS policy is a good policy that is useful for the rate-function delay analysis of other policies, yet it is neither throughput-optimal nor rate-function delay-optimal in general. Although (the modified version of) the DWM policy is both throughput-optimal and rate-function delay-optimal, it yields an impractically high complexity. Our analysis shows that our proposed DWM-n policy is rate-function delay-optimal and substantially reduces the complexity to $O(n^{2.5} \log n)$, but it is not throughput-optimal. Furthermore, we show that a simple throughput-optimal policy, the D-MWS policy, suffers from a zero rate-function.

V. Hybrid Policies

It is clear from Section IV that a policy that satisfies the sufficient conditions in Theorems 2 and 6 is both throughput-optimal and rate-function delay-optimal. It remains, however, to find such a policy with a *low complexity*. Interestingly, our carefully chosen sufficient conditions possess the following special features, which allow us to construct a *low-complexity hybrid policy that is both rate-function delay-optimal and throughput-optimal*.

- The sufficient condition for throughput optimality has a decoupling feature, in the sense that the condition can be separately verified for each individual server.
- The sufficient condition for rate-function delay optimality guarantees not only rate-function delay optimality itself, *but also that all scheduled servers for the n oldest packets satisfy the sufficient condition for throughput optimality.*

Hence, by exploiting the above useful features of our sufficient conditions, we can now develop a class of two-stage hybrid OPF-MWF policies that runs an OPF policy (focusing on the n oldest packets only) in stage 1, and runs an MWF policy in stage 2 over the remaining servers (that are not allocated in stage 1) only. We will then show that all policies in this class of hybrid OPF-MWF policies are both rate-function delay-optimal and throughput-optimal. In particular, we can find simple OPF-MWF policies with a low complexity $O(n^{2.5}\log n)$.

We now formally define the class of two-stage hybrid OPF-MWF policies.

Definition 3: A scheduling policy $\mathbf{P}$ is said to be in the class of *hybrid OPF-MWF* policies if the following conditions are satisfied under policy $\mathbf{P}$: In each time-slot t, there are two stages.

1) In stage 1, it runs an OPF policy over the n oldest packets only.
2) In stage 2, let $R(t)$ denote the set of servers that are not allocated by the OPF policy in stage 1, and let $i(j,t)$ be the index of the queue that is matched by server S_j for $j \in R(t)$ in stage 2. There exists a constant $M > 0$ such that in any time-slot t and for all $j \in R(t)$, queue $Q_{i(j,t)}$ satisfies that $W_{i(j,t)}(t) \geq Z_{r,M}(t)$ for all $r \in \Gamma_j(t)$ such that $Q_r(t) \geq M$. In other words, it runs an MWF policy over the system with the remaining servers and packets.

In the following theorem, we show that the class of OPF-MWF policies is both rate-function delay-optimal and throughput-optimal.

Theorem 9: Any hybrid OPF-MWF policy is rate-function delay-optimal under Assumptions 2 and 3, and is throughput-optimal under Assumption 1.

We provide the proof in Appendix F and give the intuition behind it as follows. In stage 1, an OPF policy not only guarantees rate-function delay optimality, *but also satisfies the sufficient condition for throughput optimality for all allocated servers in this stage*. Note that the allocated servers and packets in stage 1 will not be considered in stage 2. In stage 2, *we run an MWF policy for the remaining servers and packets only*. Hence, it ensures that the sufficient condition for throughput optimality is satisfied for the remaining servers as well. Since the allocated servers and packets in stage 1 are not touched in stage 2, the satisfaction of the sufficient condition for delay optimality is not perturbed, and the sufficient condition for throughput optimality is also satisfied.

We note that the idea of combining different policies into (heuristic) hybrid policies to improve the overall performance, is not new. However, our goal in this paper is to achieve *provable* optimality in terms of both throughput and delay. Hence, the task of designing the right hybrid policy becomes much more challenging. Furthermore, it is not necessary that all combinations of the OPF and MWF policies lead to desired hybrid policies. For example, it is unclear that the sufficient condition for throughput optimality can be satisfied if instead we run an MWF policy in stage 1 and do post-processing by applying an OPF policy in stage 2. In this case, because the servers allocated by an MWF policy in stage 1 can be reallocated in stage 2, the sufficient condition for throughput optimality may not hold any more. In contrast, our solutions exploit the special features of our carefully chosen sufficient conditions, and intelligently combine different policies in a right way, to achieve the optimal performance for both throughput and delay.

There are still many policies in the class of hybrid OPF-MWF policies. In the following, as an example, we show that the DWM-n policy combined with the D-MWS policy yields an $O(n^{2.5}\log n)$-complexity hybrid OPF-MWF policy that is both throughput-optimal and rate-function delay-optimal. Let this policy be called $\mathbf{DWM}-n-\mathbf{MWS}$ policy. Then, we present the main result of this paper in the following theorem.

Theorem 10: DWM-n-MWS policy is in the class of hybrid OPF-MWF policies and is thus both throughput-optimal and rate-function delay-optimal. Furthermore, DWM-n-MWS policy has a complexity of $O(n^{2.5}\log n)$.

To show that DWM-n-MWS is a hybrid OPF-MWF policy, it suffices to show that Condition 2) of Definition 3 is satisfied. We provide the proof in Appendix G.

VI. Simulation Results

In this section, we conduct simulations to compare the performance of the scheduling policies proposed or discussed in this paper, where the Hybrid policy we consider is DWM-n-MWS policy. We also compare the delay performance of our proposed policies along with two $O(n^2)$-complexity queue-length-based policies (i.e., using queue lengths instead of delays to calculate weights when making scheduling decisions): Queue-based Server-Side-Greedy (Q-SSG) and Q-MWS, which have been studied in [6] and [7]. We implement and simulate these policies in Java and compare the empirical probabilities that the largest HOL delay in the system in any given time-slot exceeds a constant b, i.e., $\mathbb{P}(W(0) > b)$.

For the arrival processes, we consider bursty arrivals that are driven by a two-state Markov chain and are thus correlated over time. (We obtained similar results for *i.i.d.* 0-L arrivals over time, but omit them here due to space constraints.) We adopt the same parameter settings as in [10] and [11]. For each user, there are five packet-arrivals when the Markov chain is in state 1, and no arrivals when the Markov chain 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 (as in Assumption 4) and set $q = 0.75$, and later consider more general scenarios with heterogeneous users and bursty channels that are correlated over time. We run simulations for a system with $n \in \{10, 20, \ldots, 100\}$. The simulation period lasts for 10^7 time-slots for each policy and each system.

The results are summarized in Figs. 2 and 3, where the complexity of each policy is labeled. In order to compare the rate-function $I(b)$ as defined in (2), we plot the probability over the number of channels or users, i.e., n, for a fixed value of threshold b. In Fig. 2, we compare the rate-function $I(b)$ of different scheduling policies for $b = 2$. The negative of the slope of each curve can be viewed as the rate-function for the corresponding policy. From Fig. 2, we observe that the Hybrid and DWM-n policies perform closely to DWM, and that D-MWS and Q-MWS have a zero rate-function, which supports our analytical results. Furthermore, the results show that the delay-based policies (DWM, DWM-n, and Hybrid) consistently outperform Q-SSG in terms of delay performance, despite that it has been shown through simulations that Q-SSG performs closely to a rate-function (queue-length) optimal policy [6], [7]. This provides further evidence of the fact that

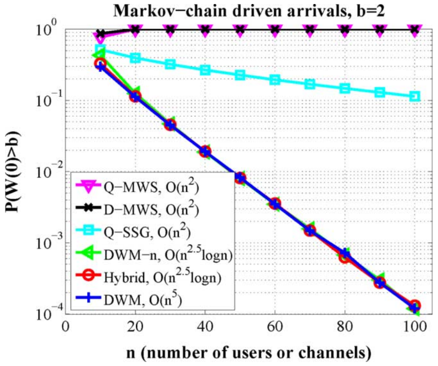

Fig. 2. Performance comparison of different scheduling policies in the case with homogeneous *i.i.d.* channels, for delay threshold $b = 2$.

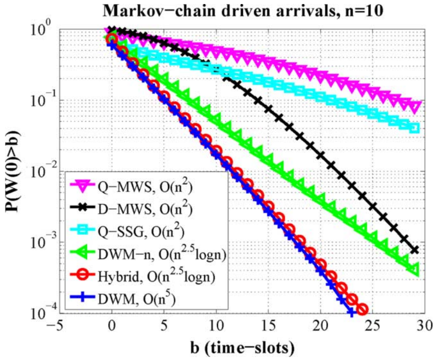

Fig. 3. Performance comparison of different scheduling policies in the case with homogeneous *i.i.d.* channels, for $n = 10$ channels/users.

good queue-length performance does not necessarily translate to good delay performance.

We also plot the probability over delay threshold b as in [6]–[8], [10], and [11] to investigate the performance of different policies when n is small. In Fig. 3, we report the results for $n = 10$ and $b \in \{0, 1, 2, \dots, 29\}$. From Fig. 3, we observe that the Hybrid policy consistently performs closely to DWM for almost all values of b that we consider, while DWM-n is worse than DWM. This is because DWM-n may not schedule all the servers, and the probability that some of the servers are kept idle can be significant when n is small.

Finally, we evaluate the performance of different scheduling policies in more realistic scenarios, where users are *heterogeneous* and channels are *correlated over time*. Specifically, we consider channels that can be modeled by a two-state Markov chain, where the channel is "ON" when the Markov chain is in state 1, and is "OFF" when the Markov chain is in state 2. This type of channel model can be viewed as a special case of the Gilbert Elliot model that is widely used for describing bursty channels. 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. Also, the delay requirements are assumed to be the same for different classes of users, i.e., we still consider the probability that the largest HOL delay exceeds a fixed threshold, without distinguishing different classes of users.

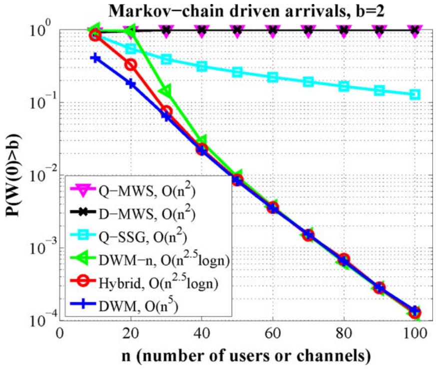

Fig. 4. Performance comparison of different scheduling policies in the case with Markov-chain driven heterogeneous channels, for delay threshold $b = 2$.

The results are summarized in Fig. 4. We observe similar results as in the previous case, where channels are *i.i.d.* in time. In particular, our low-complexity policies (DWM-n and Hybrid) again perform closely to DWM, in terms of rate-function, although the delay-violation probability is a bit smaller under DWM when n is not large (i.e., $n < 50$), which is expected. Note that in this scenario, rate-function delay-optimal policies are *not* known yet. For future work, it would be interesting to explore whether our proposed policies can achieve optimality of both throughput and delay in more general scenarios.

VII. Conclusion

In this paper, we addressed the question of designing low-complexity scheduling policies that provide optimal performance of both throughput and delay in multichannel systems. We derived simple and easy-to-verify sufficient conditions for throughput optimality and rate-function delay optimality, which allowed us to later develop a class of low-complexity hybrid policies that simultaneously achieve both throughput optimality and rate-function delay optimality.

Our work in this paper leads to many interesting questions that are worth exploring in the future. It would be interesting to know if one can further relax the sufficient conditions and design even simpler policies that can provide optimal performance for both throughput and delay. Furthermore, it would be worthwhile to analytically characterize the fundamental tradeoff between performance and complexity.

Furthermore, it is important to investigate the scheduling problem in more realistic scenarios, e.g., accounting for more general multirate channels that are correlated over time, rather than *i.i.d.* ON–OFF channels, and heterogeneous users with different statistics as well as different delay requirements. Our hope is to find efficient schedulers that can guarantee a nontrivial lower bound of the optimal rate-function if it is too hard to achieve (or prove) the optimal delay performance itself in more general scenarios.

Finally, it is interesting and important for us to understand the delay performance beyond rate-function optimality as we considered in this paper. The log-asymptotic results from the large-deviations analysis may not suffice since they do not account for the prefactor of the delay-violation probability. Therefore, a very important direction is to analyze and understand the exact delay asymptotics as well as the mean delay performance.

Appendix A
Proof of Theorem 1

We begin with stating an important property of $I_A(t,x)$ in the following lemma, which will be used in deriving the upper bound in Theorem 1. Recall that we define the quantity $I_A^+(t,x) \triangleq \lim_{y \to x^+} I_A(t,y)$.

Lemma 11: Suppose $L > 1$. For any given integer $t > 0$, and for all $x \in [0,(L-1)t)$, the limit $I_A^+(t,x) = \lim_{y \to x^+} I_A(t,y)$ exists and we have $I_A(t,x) = I_A^+(t,x)$.

Proof: Consider any given integer $t > 0$. First, note that the total number of packet arrivals to the system during an interval of t time-slots cannot exceed Lnt. Hence, we only need to consider $I_A(t,x)$ defined on $[0,(L-1)t]$. By the second part of Assumption 2, it is easy to see that $I_A(t,x)$ must be finite in $[0,(L-1)t)$. Note that $I_A(t,x)$ is a supremum (over θ) of linear functions (of x). Hence, $I_A(t,x)$ is a convex function (of x), and is thus continuous on $(0,(L-1)t)$ (i.e., the interior of $[0,(L-1)t]$) ([23, p. 68]). Furthermore, it is easy to see that $I_A(t,x)$ is monotone (nondecreasing) on $[0,(L-1)t]$ due to (3). Hence, it is not hard to show that $I_A(t,x)$ is right-continuous at the left-most point $x = 0$. Therefore, the limit $\lim_{y \to x^+} I_A(t,y)$ exists and we have $I_A(t,x) = I_A^+(t,x)$ for any $x \in [0,(L-1)t)$. ■

First, we focus on the case where $L > 1$, and consider three types of events, $\mathcal{E}_1$, $\mathcal{E}_2^c$, and $\mathcal{E}_3^c$, that imply the delay-violation 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. Furthermore, suppose that Q_1 is disconnected from all n servers in all 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 $c \in \{0,1,\ldots,b\}$ and any $t > t_{b-c}$. Recall that $t_{b-c} = \frac{b-c}{L-1}$. Then, for all $t > t_{b-c}$, we have $b - c < (L-1)t$, and thus $I_A(t,b-c) = I_A^+(t,b-c)$ from Lemma 11. Hence, for any fixed $\epsilon > 0$, there exists a $\delta > 0$ such that $I_A(t,b-c+\delta) \leq I_A^+(t,b-c) + \epsilon = I_A(t,b-c) + \epsilon$. Suppose that from time-slot $-t-b$ to $-b-1$, the total number of packet arrivals to the system is greater than or equal to $nt + n(b-c+\delta)$, and let $p_{(b-c+\delta)}$ denote the probability that this event occurs. Then, from Cramer's Theorem, we have $\lim_{n \to \infty} \frac{-1}{n} \log p_{(b-c+\delta)} = I_A(t,b-c+\delta) \leq I_A(t,b-c) + \epsilon$. Clearly, the total number of packets that are served in any time-slot is no greater than n. For any fixed δ, we have $n\delta \geq 1$ for large enough n (when $n \geq \frac{1}{\delta}$). Hence, if the above event occurs, 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 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 $-b$. Thus, 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

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

Hence, we have

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

and thus

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

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

$$\limsup_{n \to \infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq \min_{c \in \{0,1,\ldots,b\}} \left\{ \inf_{t > t_{b-c}} I_A(t,b-c) + cI_X \right\}.$$

Event $\mathcal{E}_3^c$: Consider any fixed $c \in \Psi_b$. Suppose that from time-slot $-t_{b-c} - b$ to $-b-1$, the total number of packet arrivals to the system is equal to $nt_{b-c} + n(b-c) = nLt_{b-c}$, and let $p'_{(b-c)}$ denote the probability that this event occurs. Note that the total number of packet arrivals to the system from time-slot $-t_{b-c} - b$ to $-b-1$ can never exceed nLt_{b-c}. Then, from Cramer's Theorem, we have $\lim_{n \to \infty} \frac{-1}{n} \log p'_{(b-c)} = I_A(t_{b-c}, b-c)$. Clearly, the total number of packets that can be served during the interval $[-t_{b-c} - b, -c-1]$ is no greater than $n(t_{b-c} + b - c) = nLt_{b-c}$. Suppose that there exists one queue that is disconnected from all the servers in any one time-slot in the interval $[-t_{b-c} - b, -c-1]$. Then, 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 queue Q_1 is disconnected from all the servers in a time-slot, say time-slot $-t_{b-c} - b$. This event occurs with probability $(1-q)^n = e^{-nI_X}$. Furthermore, 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 $-b$. Thus, we have $W(0) > b$ in this case. This implies $\mathcal{E}_3^c \subseteq \{W(0) > b\}$. Note that the probability that event $\mathcal{E}_3^c$ occurs can be computed as

$$\mathbb{P}(\mathcal{E}_3^c) = p'_{(b-c)} e^{-n(c+1)I_X}.$$

Hence, we have

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

and thus

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

Since the above inequality holds for any $c \in \Psi_b$, by taking the minimum over all $c \in \Psi_b$, we have, for $L > 1$

$$\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq \min_{c\in\Psi_b} \{I_A(t_{b-c}, b-c) + (c+1)I_X\}.$$

Considering events $\mathcal{E}_1$, $\mathcal{E}_2^c$, and $\mathcal{E}_3^c$, we have

$$\begin{aligned}&\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b)\\ &\leq \min\left\{(b+1)I_X, \min_{0\leq c\leq b}\left\{\inf_{t>t_{b-c}} I_A(t, b-c) + cI_X\right\},\right.\\ &\qquad\left.\min_{c\in\Psi_b}\{I_A(t_{b-c}, b-c) + (c+1)I_X\}\right\}\\ &= I_0(b).\end{aligned}$$

Next, we consider the case where $L = 1$. In this case, we only need to consider event $\mathcal{E}_1$, and we have $\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq (b+1)I_X$.

Combining both cases of $L = 1$ and $L > 1$, we have $\limsup_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \leq I_0^*(b)$. This completes our proof.

Appendix B
Proof of Theorem 2

Suppose policy $\mathbf{P}$ satisfies the sufficient condition in Theorem 2. We want to show that for any given integer threshold $b \geq 0$, the rate-function attained by policy $\mathbf{P}$ is no smaller than $I_0^*(b)$. The proof follows a similar argument as in the proof of [10, Theorem 2]. However, our proof exhibits the following key difference. In [10], the authors prove that the FBS policy can attain a certain rate-function, which, in some cases only, meets the upper bound derived in [10]. In contrast, in the following proof, by exploiting the dominance property over both the FBS policy and the perfect-matching policy in Lemma 3, we will show that the rate-function attained by policy $\mathbf{P}$ is always no smaller than the upper bound $I_0^*(b)$ that we derived in Theorem 1 and is thus optimal.

We first consider the case of $L > 1$ and want to show that the rate-function attained by policy $\mathbf{P}$ is no smaller than $I_0(b)$.

In the following proof, we will use the dominance property of policy $\mathbf{P}$ over the FBS policy and the perfect-matching policy considered in Lemma 3. We first choose the value of parameter h for FBS based on the statistics of the arrival process. We fix $\delta < \frac{2}{3}$ and $\epsilon < \frac{p}{2}$. Then, from Assumption 3, there exists a positive function $I_B(\epsilon, \delta)$ such that for all $n \geq N_B(\epsilon, \delta)$ and $t \geq T_B(\epsilon, \delta)$, we have

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

for any integer l. We then choose

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

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. Let $H = Lh$. Then, H 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.

Let $L(-b)$ be the last time before time-slot $-b$, when the backlog is empty, i.e., all the queues have a queue length of zero. Also, let $\mathcal{E}_t$ be the set of sample paths such that $L(-b) = -t - b - 1$ and $W(0) > b$ under policy $\mathbf{P}$. Then, we have

$$\mathbb{P}(W(0) > b) = \sum_{t=1}^{\infty} \mathbb{P}(\mathcal{E}_t). \qquad (6)$$

Let $\mathcal{E}_t^{\mathrm{F}}$ and $\mathcal{E}_t^{\mathrm{PM}}$ be the set of sample paths such that given $L(-b) = -t - b - 1$, the event $W(0) > b$ occurs under the FBS policy and the perfect-matching policy, respectively. Recall that policy $\mathbf{P}$ dominates both the FBS policy and the perfect-matching policy. Then, for any $t > 0$ we have

$$\mathcal{E}_t \subseteq \mathcal{E}_t^{\mathrm{F}} \cap \mathcal{E}_t^{\mathrm{PM}}. \qquad (7)$$

Recall that p is the mean arrival rate to a queue. Now, we choose any fixed real number $\hat{p} \in (p, 1)$, and fix a finite time t^* as

$$t^* \triangleq \max\left\{T_1, \left\lceil \frac{I_0(b)}{I_{BX}} \right\rceil, \max\{t_{b-c} | c \in \Psi_b\}\right\} \qquad (8)$$

where

$$T_1 \triangleq \max\left\{T_B\left(\hat{p} - p, \frac{1-\hat{p}}{6(L+2)}\right), \left\lceil \frac{6}{1-\hat{p}} \right\rceil\right\} \qquad (9)$$

and

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

The reason for defining the above value of t^* will become clear later on. Then, we apply (7) to (6) and split the summation as

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

where

$$P_1 \triangleq \sum\nolimits_{t=1}^{t^*} \mathbb{P}\left(\mathcal{E}_t^{\mathrm{F}} \cap \mathcal{E}_t^{\mathrm{PM}}\right)$$

and

$$P_2 \triangleq \sum\nolimits_{t=t^*}^{\infty} \mathbb{P}\left(\mathcal{E}_t^{\mathrm{F}} \cap \mathcal{E}_t^{\mathrm{PM}}\right).$$

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

$$P_1 \leq C_1 n^{7bH} e^{-nI_0(b)}.$$

Then, in Part 2, we show that there exists a finite $N_2 > 0$ such that for all $n \geq N_2$, we have

$$P_2 \leq 4e^{-nI_0(b)}.$$

Finally, combining both parts, we have

$$\mathbb{P}(W(0) > b) \leq (C_1 n^{7bH} + 4)e^{-nI_0(b)}$$

for all $n \geq N \triangleq \max\{N_1, N_2\}$. By taking logarithm and limit as n goes to infinity, we obtain $\liminf_{n\to\infty} \frac{-1}{n} \log \mathbb{P}(W(0) > b) \geq I_0(b)$, and thus the desired results.

The detailed proof is provided in our online technical report [14].

Appendix C Proof of Lemma 3

Suppose policy $\mathbf{P}$ satisfies the sufficient condition in Theorem 2. We first want to show that policy $\mathbf{P}$ dominates the version of the FBS policy described in Section IV-A. The proof follows a similar argument as in the proof of [10, Lemma 7].

Consider two queueing systems, $\bar{Q}_1$ and $\bar{Q}_2$, both of which have the same arrival and channel realizations. We assume that $\bar{Q}_1$ adopts policy $\mathbf{P}$ and $\bar{Q}_2$ adopts the FBS policy. Recall that the weight of a packet p in time-slot t is defined as $\hat{w}(p) = t - t_p + \frac{L+1-x_p}{(L+1)} + \frac{n+1-q(p)}{(L+1)(n+1)}$. For two packets p_1 and p_2, we say p_1 is older than p_2 if $\hat{w}(p_1) > \hat{w}(p_2)$.

Let $R_i(t)$ represent the set of packets present in the system $\bar{Q}_i$ at the end of time-slot t, for $i = 1, 2$. Then, it suffices to show that $R_1(t) \subseteq R_2(t)$ for all time t. We let $A(t)$ denote the set of packets that arrive at time t. Let $X_i(t)$ denote the set of packets that depart the system $\bar{Q}_i$ at time t, for $i = 1, 2$. Hence, we have $R_i(t+1) = (R_i(t) \cup A(t+1)) \backslash X_i(t+1)$, for $i = 1, 2$.

We then proceed with the proof by contradiction. Suppose that $R_1(t) \nsubseteq R_2(t)$ for some time t. Without loss of generality, we assume that τ is the first time such that $R_1(\tau) \nsubseteq R_2(\tau)$ occurs. Hence, there must exist a packet, say p, such that $p \in R_1(\tau)$ and $p \notin R_2(\tau)$. Because τ is the first time when such an event occurs, packet p must depart from the system $\bar{Q}_2$ in time-slot τ, i.e., $p \in X_2(\tau)$.

Let $B_i(v)$ denote the set of packets in $R_i(\tau - 1) \cup A(\tau)$ with weight greater than or equal to v, for $i = 1, 2$. Clearly, we have $B_1(v) \subseteq B_2(v)$ for all v, as $R_1(\tau - 1) \subseteq R_2(\tau - 1)$ by assumption. Since packet p is served in the system $\bar{Q}_2$ in time-slot τ, we know from the operations of FBS that all packets in $B_2(\hat{w}(p))$ must also be served in time-slot τ. This is because packet p is part of the HOL frame in time-slot τ (as packet p is served in time-slot τ), and all packets with a weight greater than $\hat{w}(p)$ must be filled to the frames with higher priority than packet p and thus should also belong to the HOL frame in time-slot τ. This further implies that in the system $\bar{Q}_1$, there exists a feasible schedule that can match all packets in $B_1(\hat{w}(p))$ since $B_1(\hat{w}(p)) \subseteq B_2(\hat{w}(p))$ and both systems have the same channel realizations.

Now, from the sufficient condition in Theorem 2, policy $\mathbf{P}$ will serve all packets in $B_1(\hat{w}(p))$, including packet p. This contradicts with the hypothesis that packet p is not served (by policy $\mathbf{P}$) in the system $\bar{Q}_1$ in time-slot τ (i.e., $p \notin R_1(\tau)$).

So far, we have shown that for any given sample path and for any value of h, by the end of any time-slot t, policy $\mathbf{P}$ has served every packet that the FBS policy has served.

Next, we want to show that policy $\mathbf{P}$ dominates the version of the perfect-matching policy described in Section IV.A. Note that in each time-slot, the packets served by the perfect-matching policy are the oldest packets in the system. The difference between FBS and the perfect-matching is the following. The HOL frame that can be served by FBS has at most Lh packets from each queue and has at most $n_0 = n - Lh$ packets from the system, while the set of packets that can be served by the perfect-matching policy has at most one packet from each queue and has at most n packets from the system. Following a similar argument as above for the FBS policy, we can show that for any given sample path, by the end of any time-slot t, policy $\mathbf{P}$ has served every packet that the perfect-matching policy has served. This completes the proof.

Appendix D Proof of Proposition 4

We first prove that DWM-n policy is an OPF policy and is thus rate-function delay-optimal. The proof follows immediately from a property of the MVM in bipartite graphs. We restate this property in the following lemma.

Lemma 12 ([16, Lemma 6]): Consider a bipartite graph, and the k heaviest vertices, for some k. If there is a matching that matches all the heaviest k vertices, then any MVM matches all of them too.

Since DWM-n policy finds an MVM in the constructed bipartite graph, Lemma 12 implies that for any $k \in \{1, 2, \ldots, n\}$, if the k oldest packets can be served by some scheduling policy, then DWM-n policy can serve these k packets as well. This completes the first part of the proof.

Next, we prove that DWM-n policy has a complexity of $O(n^{2.5} \log n)$. Note that in order to select the n oldest packets in the system, it is sufficient to sort the n^2 packets picked by DWM policy, i.e., the n oldest packets of each of the n queues, as no other packets can be among the n oldest packets in the system. The complexity of sorting n^2 packets [24] is $O(n^2 \log n)$. Given the n oldest packets in the system, DWM-n policy constructs an $n \times n$ bipartite graph and finds an MVM [16] in $O(n^{2.5} \log n)$ time. Hence, the overall complexity of DWM-n is $O(n^{2.5} \log n)$, which completes the proof.

Appendix E Proof of Proposition 5

The following simple counterexample shows that DWM-n cannot stabilize a feasible arrival rate vector and is thus not throughput-optimal in general.

Consider a system with two queues and two servers, i.e., a system with $n = 2$. We assume the *i.i.d.* ON–OFF channel model as in Assumption 4, i.e., each server is connected to each queue with probability $q \in (0, 1)$, and is disconnected otherwise. In each time-slot, a server can serve at most one packet of a queue that is connected to this server. In such a system, the optimal throughput region can be described as $\Lambda^* = \{\lambda | \lambda_1 \leq 2q, \lambda_2 \leq 2q, \text{ and } \lambda_1 + \lambda_2 \leq 2(2q - q^2)\}$, where the first two inequalities are obvious, and the last inequality is due to the following. For each of the two servers, the probability that at least one queue is connected to the server is $2q - q^2$, hence, the service each server can provide is $2q - q^2$, and the total (effective) capacity is thus $2(2q - q^2)$. Note that any arrival rate vector λ strictly inside the optimal throughput region Λ^* is feasible.

Next, we construct an arrival process as follows. Consider a frame consisting of two time-slots. In each frame, there are packet arrivals to the system with probability $p \in (0,1)$, and no arrivals otherwise. In a frame that has arrivals, there are K packet arrivals to queue Q_1 and no arrivals to queue Q_2 in the first time-slot, and there are no arrivals to queue Q_1 and K packet arrivals to queue Q_2 in the second time-slot, where we assume that $K \geq 4$. This type of arrival process yields an arrival rate vector of $\lambda^* = [\frac{pK}{2}, \frac{pK}{2}]$. It is easy to check that λ^* is feasible, if $pK \leq 4q - 2q^2$.

Now, we characterize an upper bound of the service rate under DWM-n policy. Recall that DWM-n considers only the n oldest packets in the system and maximizes the sum of the delays of the packets scheduled over these n packets, and no other packets will be scheduled. Hence, in each time-slot, DWM-n considers only the two oldest packets in the system. Consider any time-slot t_1, where $K-1$ out of the K packets arriving to queue Q_1 in the same time-slot are still waiting in the system. The other one packet could have been scheduled with a packet in Q_2, or with a packet that arrived to Q_1 earlier, or it could have been scheduled alone in a time-slot before t_1. Note that the first $K-2$ packets out of these $K-1$ packets cannot be scheduled with packets in queue Q_2, due to the operations of DWM-n. Hence, in any time-slot t_2 before these $K-1$ packets are completely evacuated, each server must serve queue Q_1 if this server is connected to queue Q_1, and no server will serve Q_2 even if this server is connected to queue Q_2, as the packets of Q_2 are not among the two oldest packets in the system in such time-slot t_2. Hence, the expected service rate for these $K-2$ packets is $2q$, and it thus takes $\frac{K-2}{2q}$ time-slots on average to evacuate the $K-2$ packets. Similarly, it takes $\frac{K-2}{2q}$ time-slots on average to evacuate such $K-2$ packets in queue Q_2. Therefore, the total service rate of the system under DWM-n is no greater than $\frac{2K}{\frac{2(K-2)}{2q}} = \frac{2qK}{K-2}$. It is clear that the system is unstable if the total arrival rate is greater than the total service rate, i.e., $pK > \frac{2qK}{K-2}$. Then, by choosing $p = \frac{17}{96}$, $q = \frac{1}{2}$ and $K = 8$, we obtain a feasible arrival rate vector λ^* that cannot be stabilized by DWM-n. This completes the proof.

APPENDIX F
PROOF OF THEOREM 9

We first show that a hybrid OPF-MWF policy is an (overall) OPF policy and is thus rate-function delay-optimal. Note that in stage 1, the operations of an OPF policy guarantee that the sufficient condition in Theorem 2 is satisfied. In stage 2, since the matched servers and packets in stage 1 will not be considered, it ensures that the operations do not perturb the satisfaction of the sufficient condition for rate-function delay optimality.

In the following, we want to show that a hybrid OPF-MWF policy is an (overall) MWF policy and is thus throughput-optimal. Let $M = n$. We want to show that the sufficient condition in Theorem 6 is satisfied, i.e., in any time-slot t and for all $j \in \{1, 2, \dots, n\}$, a hybrid OPF-MWF policy allocates server S_j to serve queue $Q_{i(j,t)}$, which satisfies that $W_{i(j,t)}(t) \geq Z_{r,n}(t)$ for all $r \in \Gamma_j(t)$ such that $Q_r(t) \geq n$.

First, we want to show that in stage 1, an OPF policy also guarantees that *all allocated servers in stage 1 satisfy the sufficient condition for throughput optimality*. Consider each server S_l such that $l \in \{1, 2, \dots, n\} \backslash R(t)$, i.e., all servers S_j that are allocated in stage 1. Then, $Q_{i(l,t)}$ is the queue served by server S_l in stage 1 of time-slot t. Since we run an OPF policy in stage 1, server S_l serves a packet among the n oldest packets in the system, and it must satisfy that $W_{i(j,t)}(t) \geq Z_{r,n}(t)$ for any $r \in \Gamma_l(t)$ such that $Q_r(t) \geq n$.

Next, consider each server S_j such that $j \in R(t)$, then $Q_{i(j,t)}$ is the queue served by server S_j in stage 2 of time-slot t. It is clear from Condition 2) of Definition 3 that $W_{i(j,t)}(t) \geq Z_{r,n}(t)$ for all $r \in \Gamma_j(t)$ such that $Q_r(t) \geq n$.

Therefore, a hybrid OPF-MWF policy is an (overall) MWF policy and is thus throughput-optimal.

APPENDIX G
PROOF OF THEOREM 10

To show that DWM-n-MWS is a hybrid OPF-MWF policy, it is sufficient to show that Condition 2) of Definition 3 is satisfied.

Given any time-slot t, consider each server S_j such that $j \in R(t)$, then $Q_{i(j,t)}$ is the queue served by server S_j in stage 2 under D-MWS. Let $M = n$. We want to show that $W_{i(j,t)}(t) \geq Z_{r,n}(t)$ for all $r \in \Gamma_j(t)$ such that $Q_r(t) \geq n$.

Let $W'_i(t)$ be the HOL delay of queue Q_i at the beginning of stage 2. Let $\Gamma'_j(t)$ denote the set of queues that are connected to server S_j and have the largest weight among the connected queues at the beginning of stage 2 of time-slot t, i.e., $\Gamma'_j(t) \triangleq \{i \in \mathcal{S}_j(t) | W'_i(t) = \max_{l \in \mathcal{S}_j(t)} W'_l(t)\}$, where $\mathcal{S}_j(t) = \{1 \leq i \leq n | C_{i,j}(t) = 1\}$. According to the operations of D-MWS, the index of queue that is served by server S_j satisfies that $i(j,t) = \min\{i | i \in \Gamma'_j(t)\}$, hence, we have $W'_{i(j,t)}(t) = W'_r(t)$ for any $r \in \Gamma'_j(t)$. This implies that $W_{i(j,t)}(t) \geq W'_{i(j,t)}(t) = W'_r(t) \geq Z_{r,n}(t)$ for any $r \in \Gamma'_j(t)$ such that $Q_r(t) \geq n$, where the last inequality is because $Q_r(t) \geq n$ and thus the HOL packet of queue Q_r at the beginning of stage 2 must not have a later position than the nth packet in queue Q_r at the beginning of time-slot t. This holds for all $j \in R(t)$ and any time-slot t. Therefore, DWM-n-MWS is a hybrid OPF-MWF policy.

Since the complexity of DWM-n and D-MWS is $O(n^{2.5} \log n)$ and $O(n^2)$, respectively, the overall complexity of DWM-n-MWS policy is $O(n^{2.5} \log n)$.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the reviewers for their valuable comments.

REFERENCES

[1] D. Shah, D. N. C. Tse, and J. N. Tsitsiklis, "Hardness of low delay network scheduling," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7810–7817, Dec. 2011.

[2] 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.

[3] A. Ganti, E. Modiano, and J. N. Tsitsiklis, "Optimal transmission scheduling in symmetric communication models with intermittent connectivity," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 998–1008, Mar. 2007.

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

[5] S. Kittipiyakul and T. Javidi, "Resource allocation in OFDMA with time-varying channel and bursty arrivals," *IEEE Commun. Lett.*, vol. 11, no. 9, pp. 708–710, Sep. 2007.

[6] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Scheduling in multi-channel wireless networks: Rate function optimality in the small-buffer regime," in *Proc. 11th ACM SIGMETRICS*, 2009, pp. 121–132.

[7] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Low-complexity scheduling algorithms for multi-channel downlink wireless networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[8] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Scheduling for small delay in multi-rate multi-channel wireless networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1251–1259.

[9] S. Bodas and T. Javidi, "Scheduling for multi-channel wireless networks: Small delay with polynomial complexity," in *Proc. IEEE WiOpt*, 2011, pp. 78–85.

[10] M. Sharma and X. Lin, "OFDM downlink scheduling for delay-optimality: Many-channel many-source asymptotics with general arrival processes," Purdue Univ., West Lafayette, IN, USA, Tech. Rep., 2011 [Online]. Available: https://engineering.purdue.edu/%7elinx/papers.html

[11] M. Sharma and X. Lin, "OFDM downlink scheduling for delay-optimality: Many-channel many-source asymptotics with general arrival processes," in *Proc. IEEE ITA*, 2011, pp. 1–10.

[12] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multihop wireless networks," *IEEE/ACM Trans. Networking*, vol. 21, no. 5, pp. 1539–1552, Oct. 2013.

[13] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, pp. 191–217, 2004.

[14] B. Ji, G. R. Gagan, X. Lin, and N. B. Shroff, "Low-complexity scheduling policies for achieving throughput and delay optimality in multi-channel wireless networks," Arxiv preprint arXiv:1301.3598, Aug. 2013 [Online]. Available: http://arxiv.org/abs/1301.3598

[15] M. Fredman and R. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.

[16] T. Spencer and E. Mayr, "Node weighted matching," *Autom., Lang. Program. Lecture Notes Comput. Sci.*, vol. 172, pp. 454–464, 1984.

[17] G. Gupta, S. Sanghavi, and N. Shroff, "Node weighted scheduling," in *Proc. 11th ACM SIGMETRICS*, 2009, pp. 97–108.

[18] J. Dai, "On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models," *Ann. Appl. Probab.*, vol. 5, no. 1, pp. 49–77, Feb. 1995.

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

[20] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[21] L. Georgiadis, M. Neely, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.

[22] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 411–424, Apr. 2005.

[23] S. P. Boyd and L. Vandenberghe*, Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[24] T. Cormen, C. Leiserson, R. Rivest, and C. Stein*, Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

Bo Ji (S'11–M'12) received the B.E. and M.E. degrees in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2012.

He is currently a Senior Member of Technical Staff with AT&T Labs, San Ramon, CA, USA. His 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, New Delhi, India, in 2005, the M.S. degree in computer science from the University of Wisconsin–Madison, Madison, WI, USA, in 2006, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, for his dissertation titled "Delay Efficient Control Policies for Wireless Networks" in 2009.

He is currently working in the telecommunications industry. His research interests are in performance modeling and optimization of communication networks and parallel computing.

Xiaojun Lin (S'02–M'05–SM'12) received the B.S. degree in electronics and information systems from Zhongshan University, Guangzhou, China, in 1994, and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2000 and 2005, respectively.

He is currently an Associate Professor of electrical and computer engineering with Purdue University. His research interests are in the analysis, control and optimization of wireless and wireline communication networks.

Dr. Lin 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 an Associate Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING and an Area Editor for *Computer Networks*, and has served as a Guest Editor for *Ad Hoc Networks*. He received the IEEE INFOCOM 2008 Best Paper Award and the 2005 Best Paper of the Year Award from the *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.

Ness B. Shroff (S'91–M'93–SM'01–F'07) received the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 1994.

He joined Purdue University, West Lafayette, IN, USA, immediately thereafter as an Assistant Professor with the School of Electrical and Computer Engineering (ECE). At Purdue, he became a Full Professor of ECE in 2003 and Director of the Center for Wireless Systems and Applications (CWSA) in 2004. In 2007, he joined The Ohio State University, Columbus, OH, USA, where he holds the Ohio Eminent Scholar Endowed Chair in Networking and Communications in the departments of ECE and Computer Science and Engineering (CSE). From 2009 to 2012, he served as a Guest Chaired Professor of wireless communications with Tsinghua University, Beijing, China. He currently holds an honorary Guest Professor with Shanghai Jiaotong University, Shanghai, China. 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 past Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING and the IEEE COMMUNICATION LETTERS. He currently serves on the Editorial Board of the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, *Computer Networks*, *IEEE Network*, and *Networking Science*. He has chaired various conferences and workshops and co-organized workshops for the NSF to chart the future of communication networks. He is an NSF CAREER awardee. He has received numerous Best Paper awards for his research—e.g., at IEEE INFOCOM 2008, IEEE INFOCOM 2006, *Journal of Communication and Networking* 2005, *Computer Networks* 2003 (two of his papers also received runner-up awards at IEEE INFOCOM 2005 and INFOCOM 2013)—and also Student Best Paper awards (from all papers whose first author is a student) at IEEE WiOPT 2013, IEEE WiOPT 2012, and IEEE IWQoS 2006.