Delay-Optimal Opportunistic Scheduling And Approximations: The Log Rule

Bilal Sadiq[†], Seung Jun Baek[*] and Gustavo de Veciana[†]
[†]Dept. of Electrical and Computer Engineering, The University of Texas at Austin
[*]College of Information and Communication, Korea University

Abstract*—This paper considers the design of opportunistic packet schedulers for users sharing a time-varying wireless channel from the performance and the robustness points of view. Firstly, for a simplified model falling in the classical Markov decision process framework where arrival and channel statistics are *known*, we numerically compute and evaluate the characteristics of mean-delay-optimal scheduling policies. The computed policies exhibit *radial sum-rate monotonicity* (RSM), i.e., when users' queues grow linearly (i.e. scaled up by a constant), the scheduler allocates service in a manner that de-emphasizes the *balancing of unequal queues* in favor of *maximizing current system throughput* (being opportunistic). This is in sharp contrast to previously proposed policies, e.g., MaxWeight and Exp rule. The latter, however, are throughput-optimal, in that without knowledge of arrival/channel statistics they achieve stability if at all feasible. To meet performance and robustness objectives, secondly, we propose a new class of policies, called the Log rule, that are radial sum-rate monotone and provably throughput optimal. Our simulations for realistic wireless channels confirm the superiority of the Log rule which achieves up to 80% reduction in mean packet delays. However, recent asymptotic analysis showed that Exp rule is optimal in terms of minimizing the asymptotic probability of max-queue overflow. In turn, in a companion paper we have shown that an RSM policy minimizes the asymptotic probability of *sum-queue* overflow. Finally, we use extensive simulations to explore the various possible design objectives for opportunistic schedulers. When users see heterogenous channels, we find that minimizing the worst asymptotic exponent across users may excessively compromise the overall delay. Our simulations show that only if perfectly tuned to the load will the Exp rule achieve low homogenous tails across users. Otherwise the Log rule achieves a 20-75% reduction in the 99^{th} percentile for most, if not all, the users. We conclude that for wireless environments, where precise resource allocation is virtually impossible, the Log rule may be more desirable for its robust and graceful degradation to unpredicted changes.

I. Introduction

This paper addresses the design of scheduling policies for a fixed number of users sharing a wireless channel. Each user's data arrives to a queue as a random stream where it awaits transmission. The wireless channel is time-varying in that the transmission rates supported for each user vary randomly over time. If the channel state is available, a policy can schedule users so as to exploit favorable channels, e.g., schedule the user which currently has the highest rate – this is referred to as opportunistic scheduling. Our objective in this paper is to evaluate the design of queue-and-channel-aware schedulers both from the point of view of performance and robustness. By robustness we informally mean a scheduler's ability to perform well for the majority of users under unpredicted/changing conditions and even transient 'overloads' relative to the desired quality of service. Furthermore, if the system becomes temporarily overloaded, it is desirable for an opportunistic scheduler to exhibit graceful degradation of service. Though there has been a substantial amount of work on the schedulers, it is still unclear whether scheduler design should be guided by the objectives like minimizing mean delay and the asymptotic probability of sum-queue overflow, or instead by objectives like minimizing the asymptotic probability of max-queue overflow. In these considerations lies the motivation for this work and our efforts to leverage analysis, where possible, and simulation to reach a better understanding of this problem.

To put our work into context, we begin by summarizing some of the key related work in this area. Among many others, [1] considers opportunistic scheduling in a setting where users' queues are *infinitely backlogged*. They identify channel-aware opportunistic scheduling policies, which maximize the sum throughput under various types of fairness constraints. The missing element in this work is the impact of queueing dynamics. Recently, [2] showed that under a *constant* load scheduling algorithms that are oblivious to queue state will incur an average delay that grows linearly in the number of users, whereas, the channel-and-queue aware schedulers can achieve an average delay that is independent of the number of users. Even before this, it was immediately recognized that, when queueing dynamics are introduced, opportunistic scheduling policies which are solely channel-aware may not be stable (i.e., keep the users' queues bounded) unless the policy is chosen carefully, e.g., using prior knowledge of mean arrival rates. For this reason, a substantial focus was placed on designing schedulers that are both channel- and queue-aware and provably *throughput-optimal*, i.e., ensure the queues' stability without any knowledge of arrival and channel statistics if indeed stability can be achieved under any policy. Except for some degenerate cases, such policies must tradeoff *maximizing current transmission rate* versus *balancing unequal queues*. Balancing queues, avoids empty queues, which enhances the ability to exploit high channel variations in the future. We will refer to this tradeoff many times in this paper. Two classes of policies known to be throughput-optimal are MaxWeight [3] (also known as Modified Largest Weighted Work/Delay First) and Exp rule [4]. Yet, stability is a weak form of performance optimality.

Thus, it is of interest to study opportunistic scheduling

This research was supported in part by AFOSR Award FA9550-07-1-0428 and NSF Award CNS-0721532.

policies that are *delay-optimal*, e.g., polices that minimize the overall average delay (per data unit) seen by the users; or policies which minimize the probability that either the sum-queue or the largest queue overflows a large buffer. These polices are harder to characterize for servers with time-varying capacity, but some results are available that we briefly discuss next.

In [5] and [6] the Longest-Connected-Queue (LCQ) and Longest-Queue-Highest-Possible-Rate (LQHPR) policies are introduced. Strong results are shown for these policies; they stochastically minimize the max and sum queue process, and thus also the max and sum queue tails and mean delay. However, in addition to assuming certain symmetry conditions on arrival and channel statistics, [5] is limited to on-off channels where only a single user can be scheduled per time slot, and [6] assumes that the scheduler can allocate service rates from the current information theoretic multiuser capacity region. In both cases, the above-mentioned tradeoff between queue balancing and throughput maximization is absent. Indeed in [5], all policies that pick a connected queue result in the same transmission rate, whereas, in case of [6], all policies that pick a service vector from the maximal points of the current capacity region, i.e. points on the max-sum-rate face, result in the same overall transmission rate. Thus one can achieve the *queue balancing* goal, without ever compromising *throughput*. Not surprisingly, in both cases the optimal policy turns out to be greedy, in that it allocates as much service rate as possible to the longest/longer queues.

The work in [7] relaxes the symmetry assumptions considering the case with heterogenous arrivals and time-varying capacities across users satisfy pathwise large deviation principles. Under these conditions the Exp rule is shown to minimize the large deviation exponent for the overflow probability for the worst case user queue. As we will see in this paper, unlike the LCQ and LQHPR policies, when such tradeoffs need to be made, policies that are optimal in asymptotic max queue overflow, are not also optimal in terms of minimizing sum-queue overflows or mean delay. In fact the policies are very different and we believe are of practical interest.

In a related work on input-queued switches, [8] explains the conjecture that de-emphasizing queue-balancing improves mean delay of Maximum-Weight Matching algorithms.

Contributions: In this paper, we begin by characterizing mean-delay optimal opportunistic schedulers for heterogenous systems where the arrival and channel statistics are known. We consider a simple model falling in the classical Markov decision process framework, where we can numerically compute the optimal scheduling policy. Our first contribution is showing that mean delay optimal policies exhibit *radial sum-rate monotonicity* (RSM), i.e., when user queues grow linearly (i.e. scaled up by a constant) the scheduler allocates service in a manner that de-emphasizes the balancing of unequal queues in favor of maximizing current system throughput (being opportunistic). This is in sharp contrast to previously proposed policies, e.g., MaxWeight and Exp rules, which, nevertheless, have the advantage of being throughput-optimal. Our second contribution is to propose a new class of policies, called the Log-rule, that are radial sum-rate monotone and provably throughput optimal. These policies are favorable both in terms of mean delay and robustness. Our simulations for a realistic wireless channels confirm the superiority of the Log-rule which achieves a 20-75% reduction in the mean packet delay. The Log rule is proposed as a practical solution but is not provably mean-delay optimal. However, in a companion paper [9] we use the approach of [7] to show that candidate RSM policy indeed minimizes the asymptotic probability of *sum-queue* overflow.

So we have at our disposal several opportunistic scheduling policies which are good for different objectives. The question remains, in designing an opportunistic scheduler should one be guided by mean or asymptotic tail results, and should one focus on individual worst case or overall system criteria?

The third contribution of this paper, we use extensive simulation to attempt to address these important objectives, and evaluate the comparative effectiveness of various policies. We make the following observations:

1. Minimizing mean delay vs asymptotic tails. Based on simulations we observe that when users see heterogenous channels, policies such as Exp rule that aim to minimize the exponential decay rate of delay distribution tail of the *worst* user may excessively compromise average delay, in some cases penalizing the tail distributions of many of the users. Our simulations show that Log rule can achieve better mean delays (overall and on a per user basis) and comparable or better distribution tails for many, if not all, the users under reasonably high loads.

2. Graceful degradation. Due to the uncertain and changing characteristics of wireless channels, precise resource allocation to meet quality of service requirements (QoS) for real-time or streaming flows is likely to be virtually impossible. As such, a desirable design objective is for a scheduler to gracefully degrade. If there is a change in the environment causing a temporary overload, then as many users as possible should meet their QoS requirements rather than all failing. Our simulation results show that Log-rule compares favorably in this regard. In a system with unpredictable heterogenous channels, there will be a wider disparity in the performance users see under the Log rule, but a substantial number of users does very well. Hence depending on the QoS objective and specific character of the change in user's channels, one could end up with no users seeing acceptable performance under the Exp rule while, say, half the users meet their QoS requirement under Log rule. Finally we note that Log rule's underlying goal of minimizing mean packet delays might be a desirable objective from the point of view maximizing throughput seen by best effort traffic.

Organization of this paper: The rest of this paper is organized as follows: In Section II, we formally state our system model and optimality criterion. In Section III, we characterize delay-optimal schedulers using dynamic programming. Section IV explores the properties of delay-optimal schedulers regarding the above mentioned tradeoff, and compares them to known throughput-optimal polices. In Section V, we give a new class of throughput-optimal polices that posses the properties of delay-optimal schedulers. Section VI presents a further discussion of scheduler design based on simulation results comparing various scheduling polices for an HDR-like downlink [10]. Section VII concludes this paper.

II. System Model

Consider the following continuous time model for scheduling n users' traffic over a shared wireless channel. Each user $i \in I = \{1, 2, ..., n\}$ is assigned a queue in which packets with independent and exponentially distributed sizes arrive as a Poisson stream with rate λ_i packets/sec. At any time t, define the (random) vector $\boldsymbol{Q}(t) = (Q_i(t) : \ i \in I) \in \mathbb{Z}_+^n$, where $Q_i(t)$ denotes the number of packets in the i^{th} queue at time t. The state of the users' wireless channels is modeled by a random vector $\boldsymbol{R}(t) = (R_i(t) : \ i \in I)$ which can take values in a finite discrete set $\mathcal{R} \subset [0, \infty)^n$, where $R_i(t)$ is the instantaneous service rate in packets/sec available to user i if the channel were dedicated to it at time t. The finiteness and discreteness of set $\mathcal{R}$ are not needed, but assumed for the ease of exposition. We assume that for all $t \neq t'$, the channel rates $\boldsymbol{R}(t)$ and $\boldsymbol{R}(t')$ are independent and have the same distribution as $\boldsymbol{R} = (R_i : \ i \in I)$, where for all $i \in I$, $E[R_i] < \infty$. We allow the channel to be split among flows at any time instant according to a stochastic vector $\boldsymbol{\sigma}(t) = (\sigma_i(t) : \ i \in I)$– recall that a stochastic vector has non-negative components that sum up to 1– in which case the service rate available to the i^{th} user at time t will be $\sigma_i(t)R_i(t)$. Let $\boldsymbol{q}(t)$ and $\boldsymbol{r}(t)$ denote realizations of $\boldsymbol{Q}(t)$ and $\boldsymbol{R}(t)$ respectively.

The problem of scheduling users for service is then to choose a vector $\boldsymbol{\sigma}(t)$ for each time instant t, such that a given optimality criterion is met. A scheduling policy is said to be static state-feedback if it chooses the vector $\boldsymbol{\sigma}(t)$ according to a *fixed* rule based solely on the *current* system state $(\boldsymbol{q}(t), \boldsymbol{r}(t))$. More precisely, a static state-feedback scheduling policy is defined as a function $\boldsymbol{f}$ which takes the system state $(\boldsymbol{q}(t), \boldsymbol{r}(t))$ at any time t into a stochastic vector $\boldsymbol{\sigma}(t)$:

$$\boldsymbol{\sigma}(t) = \boldsymbol{f}(\boldsymbol{q}(t), \boldsymbol{r}(t)) \ . \tag{1}$$

Let $\mathcal{F}$ denote the set of static state-feedback policies. Given the optimality criterion described next, Poisson arrivals, exponentially distributed packet sizes, and i.i.d. channel state vectors, there is no loss of generality in restricting our attention to the policies in $\mathcal{F}$.

A. Optimality Criterion

Consider a system initiated at $t = 0$ in state $\boldsymbol{Q}(0) = \boldsymbol{q}^0$ which evolves under scheduling policy $\boldsymbol{f}$. The expected long-run average queue for the i^{th} user is given by,

$$\overline{q}_i(\boldsymbol{f}) = \limsup_{t \to \infty} E^{\boldsymbol{f}}_{\boldsymbol{q}^0} \left[\ \frac{1}{t} \int_0^t Q_i(\tau) d\tau \ \right] \ , \tag{2}$$

where $E^{\boldsymbol{f}}_{\boldsymbol{q}}$ denotes expectation under $\boldsymbol{f}$ conditional on $\boldsymbol{Q}(0) = \boldsymbol{q}$. We define the delay-optimal scheduling policy $\boldsymbol{f}^*$ as the one which minimizes the total weighted average queue length, if it exists, for a given weight vector $\boldsymbol{w} = (w_i : \ i \in I) > 0$, i.e.,

$$\boldsymbol{f}^* \ \in \ \arg\min_{\boldsymbol{f} \in \mathcal{F}} \ \sum_{i \in I} w_i \overline{q}_i(\boldsymbol{f}) \ . \tag{3}$$

It follows from Little's Law that if the process $(\boldsymbol{Q}(t), \ t \geq 0)$ is stationary, this optimality criterion minimizes the overall (weighted) average packet delay seen by the n-users.

B. Stabilizability and Multi-User Capacity Region

Define the multi-user capacity region of a channel as the set of longrun average service rates that can be jointly offered to the $n-$users (under all possible scheduling policies). Specifically, let Φ be the set of functions from $\mathcal{R}$ to the set of stochastic vectors, i.e. $\Phi = \{\boldsymbol{\phi} : \ \sum_{j \in I} \phi_j(r) = 1 \text{ and } \phi_i(r) \geq 0, \ \forall i \in I, \ \forall r \in \mathcal{R}\}$, then the capacity region associated with the distribution of $\boldsymbol{R}$, denoted by $\mathcal{C}$, can be characterized as,

$$\mathcal{C} = \{\boldsymbol{u} : \ 0 \leq u_i \leq E[R_i \phi_i(\boldsymbol{R})] \ i \in I, \ \text{for some } \boldsymbol{\phi} \in \Phi\} \ . \tag{4}$$

The capacity region $\mathcal{C}$ is a convex polyhedron $[0, \infty)^n$ whose exact shape depends on the distribution of $\boldsymbol{R}$ [1]. Let $\mathcal{C}^v = \{\boldsymbol{v}^1, \boldsymbol{v}^2, ..., \boldsymbol{v}^L\}$ denote the set of maximal vertices of $\mathcal{C}$, where L denotes the number of vertices. For any $I' \subseteq I$, define $\mathcal{C}(I') = \{\boldsymbol{u} \in \mathcal{C} : u_i = 0, \ \forall i \notin I'\}$, where $\mathcal{C}(I')$ is the channel capacity region when the channel is shared only amongst the users in I'.

The following (restatement of) Lemma 2.1 from [11] will be used in the subsequent sections.

Lemma 1: Assume all queues are infinitely backlogged. For any $\boldsymbol{\alpha} = (\alpha_i : \ i \in I) \geq 0$, let $\boldsymbol{\beta}(\boldsymbol{\alpha}) \in \mathcal{C}$ denote the vector of average service rates seen by the queues under the policy which serves user i at time t if,

$$i \ \in \ \arg\max_{j \in I} \{ \ \alpha_j r_j(t) \ \} \ , \tag{5}$$

augmented with a tie breaking rule, then,

$$\langle \boldsymbol{\alpha}, \boldsymbol{\beta}(\boldsymbol{\alpha}) \rangle \ = \ \max_{\boldsymbol{u} \in \mathcal{C}} \langle \boldsymbol{\alpha}, \boldsymbol{u} \rangle \ . \tag{6}$$

In fact, $\boldsymbol{\alpha}$ is an outer normal vector to the capacity region $\mathcal{C}$ at point $\boldsymbol{\beta}(\boldsymbol{\alpha})$ on its boundary.

As shown in [3], the system of n-queues is stabilizable if and only if there exists a vector $\boldsymbol{\mu} = (\mu_i : \ i \in I) \in \mathcal{C}$ such that for all $i \in I$,

$$\lambda_i < \mu_i \ . \tag{7}$$

We assume that the system under consideration is stabilizable which implies that the weighted sum defined in (3) is bounded under at least one stationary policy.

III. Characterization of Delay-Optimal Policy

Consider the process $(\boldsymbol{Q}(t), \ t \geq 0)$ initiated in state $\boldsymbol{Q}(0) = \boldsymbol{q}^0$ and evolving under a policy $\boldsymbol{f}$. Then conditional on the process being in state $\boldsymbol{q}$, the i^{th} queue sees an average service rate of $\mu_i(\boldsymbol{q})$ given by,

$$\mu_i(\boldsymbol{q}) = E[\ R_i f_i(\boldsymbol{q}, \boldsymbol{R}) \ ] \ . \tag{8}$$

By definition of $\mathcal{C}$ in (4), the average service rate vector $\boldsymbol{\mu}(\boldsymbol{q}) = (\mu_i(\boldsymbol{q}) : i \in I)$ lies in $\mathcal{C}$. We assume that over an epoch, each queue $i \in I$ is served constantly at rate $\mu_i(\boldsymbol{q})$, thus the set of n queues see state-dependent service rates chosen from $\mathcal{C}$. A rigorous justification of this can be found in [12] and relies on packet or file dynamics that are slow relative to channel variations, where the latter can be averaged. A similar assumption is made in [11] to obtain processor-sharing queueing model for a slotted time system where a packet (or file) typically takes many slots to process while the channel can change from slot to slot. Note that strictly speaking, it is shown in [13] that analysis under the assumption of infinitely fast channel variations leads to optimistic flow-level performance estimates.

Under these assumptions, the scheduling problem of finding the right function $\boldsymbol{f}(\boldsymbol{q},.)$ for each $\boldsymbol{q}$ such that the total (weighted) average queue length is minimized (see (3)), is that of finding the right service rate vector $\boldsymbol{\mu}(\boldsymbol{q}) \in \mathcal{C}$ for each $\boldsymbol{q}$. Using this, we re-define a scheduling policy as a function $\boldsymbol{\mu} : \mathbb{Z}_+^n \to \mathcal{C}$ that takes a queue state vector in $\mathbb{Z}_+^n$ to a service rate vector in $\mathcal{C}$, where $\boldsymbol{\mu}$ relates to $\boldsymbol{f}$ through (8).

Under a fixed policy $\boldsymbol{\mu}$, the process $(\boldsymbol{Q}(t),\ t \geq 0)$ forms a Markov chain on $\mathbb{Z}_+^n$ with state-dependent transition rates. For convenience, we shall uniformize $\boldsymbol{Q}(t)$. For any $\boldsymbol{q} \in \mathbb{Z}_+^n$, let $\boldsymbol{A}_i\boldsymbol{q} = \boldsymbol{q} + e^i$ and $\boldsymbol{D}_i\boldsymbol{q} = (\boldsymbol{q} - e^i)^+$, where $\boldsymbol{e}^i$ is the i^{th} standard basis element and $\boldsymbol{q}^+ = (\boldsymbol{y} : y_i = \max\{0, q_i\})$. Let $\gamma = |\boldsymbol{\lambda}| + \max_{\boldsymbol{u}\in\mathcal{C}} |\boldsymbol{u}|$, where $|\cdot|$ denotes L_1 norm. Let τ_k denote the (random) time of the k^{th} transition of $\boldsymbol{Q}(t)$ and $\tau_0 = 0$. Also, let $\boldsymbol{Q}_k = \boldsymbol{Q}(\tau_k^+)$. Then under policy $\boldsymbol{\mu}$, the process $\boldsymbol{Q}(t)$ can be viewed as having a state-independent event rate of γ (i.e. $(\tau_{k+1} - \tau_k) \sim \exp(\gamma)$) and transition probabilities given by, for all $i \in I$,

$$
\begin{aligned}
P(\ \boldsymbol{Q}_{k+1} = \boldsymbol{A}_i\boldsymbol{q} | \boldsymbol{Q}_k = \boldsymbol{q}\ ) &= \frac{\lambda_i}{\gamma}\ , \\
P(\ \boldsymbol{Q}_{k+1} = \boldsymbol{D}_i\boldsymbol{q} | \boldsymbol{Q}_k = \boldsymbol{q}\ ) &= \frac{\mu_i(\boldsymbol{q})}{\gamma}\ , \\
P(\ \boldsymbol{Q}_{k+1} = \boldsymbol{q} | \boldsymbol{Q}_k = \boldsymbol{q}\ ) &= 1 - \frac{|\boldsymbol{\lambda}| + |\boldsymbol{\mu}(\boldsymbol{q})|}{\gamma}\ .
\end{aligned}
$$

Define the cost under policy $\boldsymbol{\mu}$ over $[0, \tau_k)$ when starting in state $\boldsymbol{q}$ as $E_{\boldsymbol{q}}^{\boldsymbol{\mu}}[\ \int_0^{\tau_k} \boldsymbol{w}'\boldsymbol{Q}(t)dt\ ]$ (where $\boldsymbol{w}'$ denotes the transpose of $\boldsymbol{w}$), which, ignoring a constant multiplier γ^{-1}, can be shown to be equal to,

$$
V_k^{\boldsymbol{\mu}}(\boldsymbol{q}) = E_{\boldsymbol{q}}^{\boldsymbol{\mu}}[\ \sum_{l=0}^{k-1} \boldsymbol{w}'\boldsymbol{Q}_l\ ]\ . \tag{9}
$$

Likewise, the average cost under policy $\boldsymbol{\mu}$, when starting in state $\boldsymbol{q}$, is given by,

$$
J^{\boldsymbol{\mu}}(\boldsymbol{q}) = \limsup_{k\to\infty} \frac{1}{k} V_k^{\boldsymbol{\mu}}(\boldsymbol{q})\ . \tag{10}
$$

The optimality criterion given in (3) seeks to minimize this average cost. The problem of finding the minimum average cost and an optimal policy fits the classical dynamic programming framework (e.g. see [14]). Thus the minimum average cost over all policies (denoted by J^*) is well defined, independent of the starting state, and together with a bias function $h : \mathbb{Z}_+^n \to [0, \infty)$, which is unique up to an additive constant, satisfies Bellman's equation, i.e., for all $\boldsymbol{q} \in \mathbb{Z}_+^n$,

$$
\begin{aligned}
J^* &= \min_{\boldsymbol{\mu}} \Big\{ \boldsymbol{w}'\boldsymbol{q} + E^{\boldsymbol{\mu}}[\ h(\boldsymbol{Q}_{k+1}) - h(\boldsymbol{Q}_k)\ | \boldsymbol{Q}_k = \boldsymbol{q}\ ] \Big\}\ , \\
&= \min_{\boldsymbol{u}\in\mathcal{C}} \Big\{ \boldsymbol{w}'\boldsymbol{q} + \sum_{i=0}^{n} \frac{\lambda_i}{\gamma} \big(h(\boldsymbol{A}_i\boldsymbol{q}) - h(\boldsymbol{q})\big)\ + \\
&\qquad \frac{u_i}{\gamma}\big(h(\boldsymbol{D}_i\boldsymbol{q}) - h(\boldsymbol{q})\big) \Big\}\ .
\end{aligned} \tag{11}
$$

Moreover, let $\boldsymbol{\Delta}h(\boldsymbol{q}) = \big(h(\boldsymbol{q}) - h(\boldsymbol{D}_i\boldsymbol{q}) :\ i \in I\big)$, and define a policy $\boldsymbol{\mu}^*$ as,

$$
\boldsymbol{\mu}^*(\boldsymbol{q}) \in \arg\max_{\boldsymbol{u}\in\mathcal{C}}\ \langle \boldsymbol{u}, \boldsymbol{\Delta}h(\boldsymbol{q})\rangle, \tag{12}
$$

i.e., for each $\boldsymbol{q} \in \mathbb{Z}_+^n$, the policy $\boldsymbol{\mu}^*$ picks a service rate vector from $\mathcal{C}$ which achieves the minimum in (11). Then $\boldsymbol{\mu}^*$ is an optimal policy achieving the minimum average cost J^*. If the minimum in (12) is achieved by more than one service rate vector, then $\boldsymbol{\mu}^*(\boldsymbol{q})$ can map to any of those. The following lemma states the optimal scheduling decisions in *time* (see (1)), and follows from Lemma 1 (compare (6) and (12)):

Lemma 2: The following policy achieves the minimum average cost (see (3) and (10)): when the system is in state $(\boldsymbol{Q}, \boldsymbol{R}) = (\boldsymbol{q}, \boldsymbol{r})$ (see (1)), choose a stochastic vector $\boldsymbol{\sigma}$ that satisfies

$$
\sigma_i = 1 \textit{ for some } i \in \arg\max_{j\in I}\{\ \Delta_j h(\boldsymbol{q})\ r_j\ \}\ , \tag{13}
$$

where h is a bias function satisfying Bellman's equation (11).

Lemma 2 and (12) relate the tradeoff mentioned in Section I to the geometry of vector field $\boldsymbol{\Delta}h$ associated with the bias function. We explore this tradeoff in the next section where we use relative value iteration (see, e.g., [15]) to numerically compute h and J^*.

IV. Radial Sum-rate Monotonicity: Comparing the Optimal Policy with Known Heuristics

In this section, we investigate how delay-optimal schedulers, as well as throughput-optimal policies such as MaxWeight and Exp rule, tradeoff current transmission rate versus balancing queues. Specifically, we consider how the service rate vector chosen by each policy changes as the queues grow proportionally from a state $\boldsymbol{q} \in \mathbb{Z}_+^n$ to a state $\theta\boldsymbol{q} \in \mathbb{Z}_+^n$ for $\theta > 1$. Note that $\boldsymbol{q}$ and $\theta\boldsymbol{q}$ lie on a line in $\mathbb{R}_+^n$ that passes through origin. For any $\boldsymbol{q}$, let $I_{\boldsymbol{q}} = \{i \in I : q_i \neq 0\}$. We begin by defining an interesting property which we refer to as *radial sum-rate monotonicity*.

Definition 1: Given a weight vector $\boldsymbol{w}$, we say a scheduling policy $\boldsymbol{\mu}$ is *radially sum-rate monotone* with respect to vector $\boldsymbol{w}$ if for any $\boldsymbol{q}$ and scalar θ such that $\theta\boldsymbol{q} \in \mathbb{Z}_+^n$, the total weighted service rate, $\langle \boldsymbol{w}, \boldsymbol{\mu}(\theta\boldsymbol{q})\rangle$, is increasing function of θ, and $\lim_{\theta\to\infty}\langle \boldsymbol{w}, \boldsymbol{\mu}(\theta\boldsymbol{q})\rangle = \max(\langle \boldsymbol{w}, \boldsymbol{u}\rangle : \boldsymbol{u} \in \mathcal{C}(I_{\boldsymbol{q}}))$.

Hence, as the queues grow proportionally, a radially sum-rate monotone policy allocates service rates in a manner that de-emphasizes queue-balancing in favor of increasing the

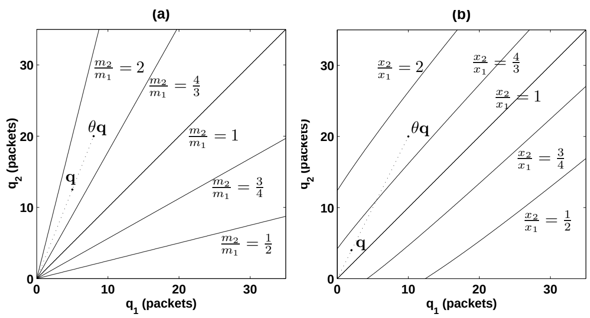

Fig. 2. Curves along which the direction is held constant by the vector field (a) $\boldsymbol{m}$ with $a_i = 1\ \forall i \in I,\ \alpha = 0.5$, (b) $\boldsymbol{x}$ with $a_i = 0.1,\ b_i = 1\ \forall i \in I,\ c = 1,\ \eta = 0.5$

total weighted service rate (with respect to weight vector $\boldsymbol{w}$). Another useful and natural property, called *transition monotonicity* [16], implies that for all $i \in I$, $\mu_i(\boldsymbol{q}) \le \mu_i(\boldsymbol{A}_i\boldsymbol{q})$ $\big($and for all $j \neq i$, $\mu_j(\boldsymbol{q}) \ge \mu_j(\boldsymbol{A}_i\boldsymbol{q})\big)$.

A. The Tradeoff Under Delay-Optimal Schedulers

Since the capacity region $\mathcal{C}$ is a polyhedron, instead of searching over the entire region for the maximum in (12), it suffices to only consider the vertices of $\mathcal{C}$,

$$\boldsymbol{\mu}^*(\boldsymbol{q}) \in \arg\max_{\boldsymbol{u}\in\mathcal{C}^v} \langle \boldsymbol{u}, \boldsymbol{\Delta} h(\boldsymbol{q})\rangle \ . \tag{14}$$

Hence, the optimal policy partitions the state space $\mathbb{Z}_+^n$ into at most L non-empty sets $\mathcal{S}^1, \mathcal{S}^2, ..., \mathcal{S}^L$, each of which is associated with a distinct vertex, i.e. $\mathcal{S}^l = \{\boldsymbol{q} : \boldsymbol{\mu}^*(\boldsymbol{q}) = \boldsymbol{v}^l\}$. In each partition $\mathcal{S}^l$, the scheduler tries to *push* the queue process $\boldsymbol{Q}_k$ along vector $\boldsymbol{\lambda} - \boldsymbol{v}^l$. Fig. 1 shows the optimal policy's partitions of $\mathbb{Z}_+^n$ for a two user system with weight vector $\boldsymbol{w} = (1, 1)$ under three different arrival vectors. The first plot shows the hypothetical 2-user capacity region and arrival vectors considered. The second plot depicts the partitions for $\boldsymbol{\lambda} = (0.25\ 0.25)$ packets/sec. The third plot exhibits a more pronounced radial sum-rate monotonicity when arrival rate is increased to $\boldsymbol{\lambda} = (0.4\ 0.4)$ packets/sec, and the last plot is intended to exhibit the warping effect on the partitions resulting from asymmetric arrival rates $\boldsymbol{\lambda} = (0.4\ 0.25)$ packets/sec.

For the optimal policy to be RSM we must have that as $\theta \to \infty$ such that $\theta\boldsymbol{q} \in \mathbb{Z}_+^n$ and $q_i > 0,\ q_j > 0$, we have $\frac{\Delta_i h(\theta\boldsymbol{q})}{\Delta_j h(\theta\boldsymbol{q})} \to \frac{w_i}{w_j}$ monotonically (i.e. $\lim_{\theta\to\infty} \boldsymbol{\Delta} h(\theta\boldsymbol{q}) \propto (w_i \mathbb{1}_{\{q_i>0\}} :\ i \in I)$). By computing the bias function h and the optimal scheduling policy for various arrival vectors and capacity regions, it can easily be seen that the optimal policy is RSM. An intuitive explanation for this is as follows: firstly, the cost incurred per unit time in state $\theta\boldsymbol{q}$ for $\theta > 1$ is more than the cost incurred per unit time in state $\boldsymbol{q}$; secondly, the state $\theta\boldsymbol{q}$ is *farther* from any axis than the state $\boldsymbol{q}$. Both these factors suggest that a scheduler can indeed increase current throughput and decrease emphasis on queue balancing. So when in state $\theta\boldsymbol{q}$, the optimal policy is relatively less "willing" to compromise current throughput in order to balance unequal queues.

Weighted Max-Rate Horn: Consider the partitions $\mathcal{S}^4$ and $\mathcal{S}^5$, i.e. partitions corresponding to the vertices of region $\mathcal{C}$ that have the largest projection along vector $\boldsymbol{w}$. Under the optimal policy, union of these partitions is shaped like a French horn (referred to as weighted max-rate horn). As we shall see next, under the Exp rule (with appropriately chosen constants), the union of the same partitions is shaped like a funnel, rapidly transforming into a cylinder, whereas, under MaxWeight, all partitions are simply cones.

B. The Tradeoffs Under MaxWeight and Exp Rule

The MaxWeight and the Exp rule policies can also be expressed in a similar form as (12). These policies replace $\boldsymbol{\Delta} h$ with a suitable vector field on $\mathbb{Z}_+^n$ such that the system is stable for any stabilizable $\boldsymbol{\lambda}$. Hence the tradeoff under each policy can be investigated by considering how the vector fields change direction as queues grow proportionally.

MaxWeight policies [3] can be defined as follows: when the system is in state $(\boldsymbol{Q}, \boldsymbol{R}) = (\boldsymbol{q}, \boldsymbol{r})$, choose a stochastic vector $\boldsymbol{\sigma}$ that satisfies

$$\sigma_i = 1 \text{ for some } i \in \arg\max_{j\in I}\{\ m_j(\boldsymbol{q})\ r_j\ \}\ ,$$

where $m_i(\boldsymbol{q})$ is the i^{th} component of $\boldsymbol{m}(\boldsymbol{q}) = (b_i q_i^{\alpha} :\ i \in I)$, for any fixed positive b_i's and α. Equivalently, when the queue state is $\boldsymbol{q}$, the policy uses a service rate vector $\boldsymbol{\mu}^M(\boldsymbol{q})$ given by,

$$\boldsymbol{\mu}^M(\boldsymbol{q}) \in \arg\max_{\boldsymbol{u}\in\mathcal{C}^v} \langle \boldsymbol{u}, \boldsymbol{m}(\boldsymbol{q})\rangle \ . \tag{15}$$

Similarly, the Exp rule [4] is given by,

$$\boldsymbol{\mu}^X(\boldsymbol{q}) = \arg\max_{\boldsymbol{u}\in\mathcal{C}^v} \langle \boldsymbol{u}, \boldsymbol{x}(\boldsymbol{q})\rangle \ , \tag{16}$$

where,

$$\boldsymbol{x}(\boldsymbol{q}) = (b_i \exp(\frac{a_i q_i}{c + (n^{-1}\sum_{j\in I} a_j q_j)^{\eta}}) :\ i \in I)\ ,$$

for any fixed positive a_i's, b_i's, c, and $0 < \eta < 1$.

While both the MaxWeight and the Exp rule are transition monotone, neither is radially sum-rate monotone. For $n = 2$ and extending the domain of $\boldsymbol{m}$ and $\boldsymbol{x}$ to $\mathbb{R}_+^n$, Fig. 2 shows the curves in $\mathbb{R}_+^2$ along which the vector fields $\boldsymbol{m}$ and $\boldsymbol{x}$ hold their direction (curves like these form the boundaries of partitions, called the switching curves). The vector field $\boldsymbol{m}$ is homogeneous, hence the service rate allocation under MaxWeight is invariant as the queues grow from state $\boldsymbol{q}$ to state $\theta\boldsymbol{q}$. By contrast, in the case of the Exp rule (with $\boldsymbol{b}$ set to $\boldsymbol{w}$), the total weighted service rate $\langle \boldsymbol{w}, \boldsymbol{\mu}^X(\theta\boldsymbol{q})\rangle$ decreases with θ and the emphasis shifts to queue-balancing, so much so that as $\theta \to \infty$, only the longest weighted queue(s) receives service.

V. Improved Throughput-Optimal Policies

We begin this section with a sufficiency theorem regarding throughput-optimal policies.

Theorem 1: Let $\boldsymbol{g} : \mathbb{R}_+^n \to \mathbb{R}_+^n$ be a gradient field (i.e. $\boldsymbol{g} = \boldsymbol{\nabla} G$ for some $G : \mathbb{R}_+^n \to \mathbb{R}$). Moreover, suppose $\boldsymbol{g}$ is

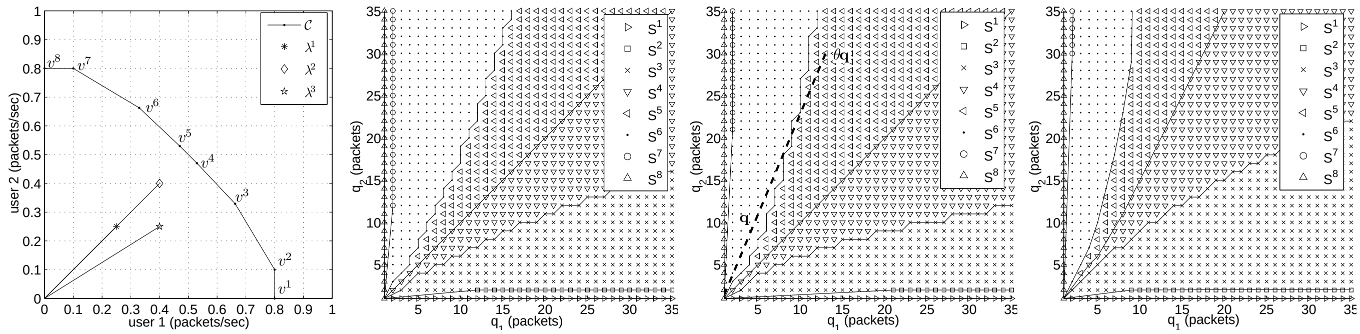

Fig. 1. Partitions under the optimal policy: from left to right, (a) 2-user capacity region. Remaining three figures show partitions corresponding to (b) arrival vector $\lambda^1 = (0.25\ 0.25)$ (c) arrival vector $\lambda^2 = (0.4\ 0.4)$ (d) arrival vector $\lambda^3 = (0.4\ 0.25)$

differentiable on $\mathbb{R}_+^n$ and for all $i \in I$, $\boldsymbol{g}$ satisfies,

$$\lim_{\boldsymbol{y}\to\infty : y_i = 0} \frac{g_i(\boldsymbol{y})}{|\boldsymbol{g}(\boldsymbol{y})|} = 0 , \tag{17}$$

$$\lim_{\boldsymbol{y}\to\infty} \frac{\partial g_i(\boldsymbol{y})/\partial y_i}{|\boldsymbol{g}(\boldsymbol{y})|} = 0 , \tag{18}$$

and for some $\epsilon > 0$, $|\boldsymbol{g}(\boldsymbol{y})| > \epsilon$ for all $\boldsymbol{y}$ outside a compact subset of $\mathbb{R}_+^n$, then the following policy $\hat{\boldsymbol{\mu}}$ is throughput-optimal: for all $\boldsymbol{q} \in \mathbb{Z}_+^n$,

$$\hat{\boldsymbol{\mu}}(\boldsymbol{q}) = \arg\max_{\boldsymbol{u}\in\mathcal{C}} \langle \boldsymbol{u}, \boldsymbol{g}(\boldsymbol{q})\rangle . \tag{19}$$

Remark: The condition that $\boldsymbol{g}$ be a gradient field and $|\boldsymbol{g}(\boldsymbol{y})| > \epsilon$ outside a compact set, is used to establish the existence of a potential (Lyapunov) function G such that $\boldsymbol{\nabla} G = \boldsymbol{g}$. Condition (17) is needed to ensure that when queue state vector $\boldsymbol{q}$ is large, the policy given by $\boldsymbol{g}$ is work-conserving, i.e., it does not allocate any service rate to an empty queue at the cost of non-empty queues. Condition (18) is used to ensure that the Hessian of G can be dominated by its gradient in the Taylor expansion (see proof). See Appendix for proof.

Examples: Examples of functions that satisfy the conditions of Theorem 1 are $\boldsymbol{m}(\cdot)$ with its domain extended to $\mathbb{R}_+^n$, $\boldsymbol{g}(\boldsymbol{y}) = (\exp(y_i^\alpha) : i \in I)$ for $0 < \alpha < 1$, and $\boldsymbol{g}(\boldsymbol{y}) = (\log(1 + \log(1 + y_i)) : i \in I)$, indicating that a throughput-optimal policy can exhibit anywhere from sub-logarithmical to almost exponential sensitivity to changes in queue lengths.

A. The Log Rule

In this section we consider a class of schedulers satisfying Theorem 1, which we refer to as the Log rule.

Definition 2: Arbitrarily fix $\boldsymbol{a} = (a_i : i \in I) > 0$, $\boldsymbol{b} = (b_i : i \in I) > 0$, and $c \geq 1$. For all $\boldsymbol{y} \in \mathbb{R}_+^n$, let $\boldsymbol{g}^L(\boldsymbol{y}) = (g_i^L(\boldsymbol{y}) : i \in I)$, where $g_i^L(\boldsymbol{y}) = b_i \log(c + a_i y_i)$. When the system is in state $(\boldsymbol{Q}, \boldsymbol{R}) = (\boldsymbol{q}, \boldsymbol{r})$ (see (1)), choose a stochastic vector $\boldsymbol{\sigma}$ that satisfies,

$$\sigma_i = 0 \text{ if } i \notin \arg\max_{j\in I}\{\ g_j^L(\boldsymbol{q})\ r_j\ \} . \tag{20}$$

Theorem 2: The Log rule is radial sum-rate monotone w.r.t. weight vector b (with $c = 1$) and throughput-optimal.

Proof: First the throughput optimality: let $\boldsymbol{\mu}^L(\boldsymbol{q}) = (\mu_i^L(\boldsymbol{q}),\ i \in I)$ denote the vector of service rates (under the Log-Rule) seen by the queues when $\boldsymbol{Q}(t) = \boldsymbol{q}$, then

$$\mu_i^L(\boldsymbol{q}) = E[\ R_i g_i^L(\boldsymbol{q}, \boldsymbol{R})\ ] , \tag{21}$$

where $\boldsymbol{g}^L(\boldsymbol{q}, \boldsymbol{r})$ denotes the stochastic vector chosen by the Log-Rule (i.e. satisfying (20)) in state $(\boldsymbol{q}, \boldsymbol{r})$. By (20) and Lemma 1, $\langle \boldsymbol{g}^L(\boldsymbol{q}), \boldsymbol{\mu}^L(\boldsymbol{q})\rangle = \max_{\boldsymbol{u}\in\mathcal{C}} \langle \boldsymbol{g}^L(\boldsymbol{q}), \boldsymbol{u}\rangle$. Moreover, the function $\boldsymbol{g}^L$ satisfies the conditions of Theorem 1, and the throughput-optimality of the Log-Rule follows. To verify the radial sum-rate monotonicity of the Log rule we note that for any $\boldsymbol{q} \in \mathbb{Z}_+^n$ such that $0 < a_i q_i < a_j q_j$, we have $\frac{g_i^L(\theta\boldsymbol{q})}{g_j^L(\theta\boldsymbol{q})} \nearrow \frac{b_i}{b_j}$ as $\theta \to \infty$. ∎

Note that as $\theta \to 0$, we have $\frac{g_i^L(\theta\boldsymbol{q})}{g_j^L(\theta\boldsymbol{q})} \to \frac{b_i}{b_j}$, i.e., close to origin in $\mathbb{Z}_+^n$, Log rule behaves similar to MaxWeight with $\alpha = 1$, whereas, radially far away from origin (as $\theta \to \infty$), $\boldsymbol{g}^L(\theta\boldsymbol{q})$ becomes parallel to the vector $(b_i \mathbb{1}_{\{q_i > 0\}},\ i \in I)$ and the Log rule ignores queue-balancing in favor of maximizing the total weighted service rate, $\langle \boldsymbol{b}, \boldsymbol{\mu}^L(\theta\boldsymbol{q})$.

Fig. 3 shows the curves along which the direction of the gradient field $\boldsymbol{g}^L$ is constant, (curves like these form the switching curves and define partitions of the queue state-space). A good choice for w_i (hence b_i) is $1/E[R_i]$, as suggested for the Exp rule in [17]. The line $\{\boldsymbol{q} \in \mathbb{Z}_+^n : a_i q_i = a_j q_j\ \forall i, j \in I\}$ defines the axis of the weighted max-rate horn, whereas, the magnitude of the vector $\boldsymbol{a}$ controls the width of the horn (or convergence of the above limits.) Increasing the magnitude of $\boldsymbol{a}$ widens the horn and reduces the emphasis of Log rule on balancing user queues (this is opposite to the role this parameter plays in the Exp rule). By choosing $c > 1$, the Log rule can be made to behave similar to the Exp rule, instead of MaxWeight with $\alpha = 1$, near the origin in $\mathbb{Z}_+^n$.

Asymptotic Probability of Sum-queue Overflow under the Log Rule: Due to space constraints we have relegated the proof of the Log rule's asymptotic optimality to a companion paper [9]. By leveraging the refined sample path large deviations principle, recently introduced in [7] to study non-homogenous schedulers like the Exp rule and the Log rule, we are able to show that a Log-rule-like policy satisfying radial sum-rate monotonicity (w.r.t vector $(1, 1)$) indeed minimizes

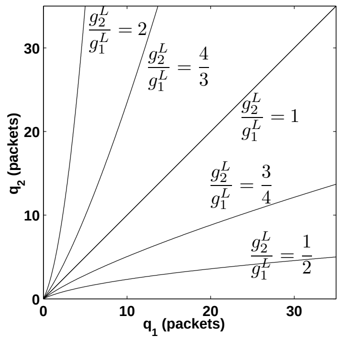

Fig. 3. Partitions under the Log-Rule with $a_i = 1,\ b_i = 1\ \forall i \in I,\ c = 1$

the asymptotic probability of sum queue overflow, i.e.,

$$\limsup_{n\to\infty} \frac{1}{n} \log P\big(|\boldsymbol{Q}(0)| > n\big) \ ,$$

where $P(.)$ denotes the stationary distribution of the Markov chain $\boldsymbol{Q}$ under a stable scheduling policy. Leaving to basic questions of which design objective is appropriate in designing opportunistic schedulers, and whether the asymptotic results are sufficiently accurate to dictate which class of scheduler should be used. We consider these questions in the next section.

VI. Evaluating Opportunistic Scheduler Design Objectives – Simulations

In this section we discuss a simulation-based evaluation of opportunistic schedulers from various perspectives:

(a) performance, including mean packet delays and 99^{th} percentile delays of individual users as well the overall system;

(b) sensitivity to both scheduler parameters and channel characteristics;

(c) and graceful degradation, in terms of the fraction of users that meet QoS objectives under overloads.

Note we consider a system as overloaded if it can no longer meet users' QoS requirements, this might be due to a change in the channel characteristics, due to mobility etc. These perspectives are clearly interrelated yet for clarity we discuss them separately.

A. Simulation Model and Operational Scenarios

We choose an HDR-like wireless downlink [10] to compare various scheduling rules, namely Log rule, MaxWeight, and Exp rule. Performance comparisons for an HDR downlink under Proportional Fair scheduling, MaxWeight, and Exp rule were presented in [17], and showed the Exp rule to be superior to the others. Note that the HDR downlink model differs from the system model presented in Section II, however, we choose this as our simulation model to demonstrate the practical significance of our proposed scheduling rule and allow comparison with other simulations and theoretical work in the literature. Thus, instead of i.i.d. channel, continuous time scheduling, and Poisson arrivals with exponentially distributed packets sizes, here we assume that channels are *correlated* over time, scheduling decisions are made once in each time slot of duration 1.67 ms, and each user's packets are 1Kb and arrive as i.i.d. Bernoulli processes.

User i	1	2	3	4	5	6
$E[R_i]$ kbps	572.8	392.1	304.6	250.1	215.1	187.9
User i	7	8	9	10	11	12
$E[R_i]$ kbps	167.6	151.3	138.0	127.2	117.1	109.6

TABLE I
Mean data rate supported by wireless channel of each user

LOG	EXP	MW
$b_i = \frac{1}{E[R_i]}$ $a_i = 10$ $c = 10$	$b_i = \frac{1}{E[R_i]}$ $a_i = 0.05,$ $c = 1, \eta = 0.5$	$b_i = \frac{1}{E[R_i]}$ $\alpha = 1$

TABLE II
Parameters used for each scheduling policy

We consider $n = 12$ heterogenous users connected to a single access point. The locations of the n users are taken to be uniformly distributed in a circular cell; Table I gives the mean data rate $E[R_i]$ in bits/sec that the wireless channel of each user can support. Moreover, the wireless link between the access point and each user is taken as an independent Rayleigh fading channel with a Doppler frequency of 18 Hz. Specifically, in any time slot $t \in \mathbb{Z}$, the channel state (rate supported by the channel) of i^{th} user is given by,

$$R_i(t) = \text{BW} \times \log_2\left(1 + \text{SINR}_i(t)\right) \quad \text{bits/sec}$$

and SINR (signal-to-interference-plus-noise ratio) is assumed to hold its value over the duration of the time slot. During each time slot, data is transmitted to a single user who is selected according to the scheduling policy. If user i is selected in time slot t, then (at most) $1.67\text{ms} \times R_i(t)$ bits are transmitted from its queue.

Due to space constraints we are only able to present simulation results for five operational scenarios. In the first three scenarios users see heterogenous channels but have homogenous traffic with *low* $\boldsymbol{\lambda}^{(s,l)}$, *medium* $\boldsymbol{\lambda}^{(s,m)}$, or *high* $\boldsymbol{\lambda}^{(s,h)}$ rates given by,

$$\begin{aligned} \lambda_i^{(s,m)} &= 2.3 \times \left(\sum_{j=1}^{n} \frac{1}{E[R_j]}\right)^{-1} \times \frac{1}{1Kb} \quad \text{packets/sec}, \\ \lambda_i^{(s,l)} &= 0.98 \times \lambda_i^{(s,m)}, \quad \lambda_i^{(s,h)} = 1.02 \times \lambda_i^{(s,m)}. \end{aligned}$$

In words, for the medium case a user's arrival rate is 2.3 times higher than that is stabilizable by a non-opportunistic scheduler; the low and the high arrival rates are respectively 2% lower and higher than the medium. Fig. 4-(a) and (b) show performance results under these three homogenous load scenarios – see caption for detailed explanation. In the fourth scenario the traffic load is kept *low* but User 7 (see Table I) is moved to the edge of cell, which increases the system load. Fig. 4-(c) exhibits the results for this case. For the fifth scenario, users have heterogenous arrival rates given by,

$$\lambda_i = 2.35 \times \frac{E[R_i]}{n} \times \frac{1}{1Kb} \quad \text{packets/sec}$$

i.e., arrival rate vector $\boldsymbol{\lambda}$ is proportional to the mean channel

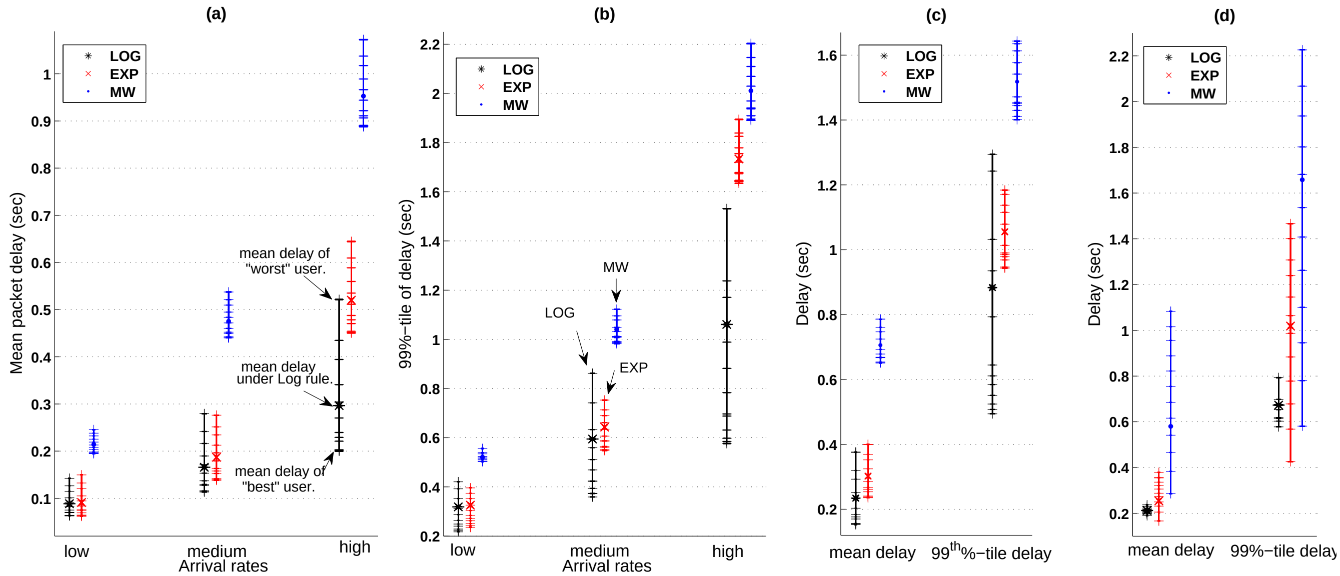

Fig. 4. Simulation-based performance comparisons for three opportunistic scheduling policies, Log Rule, Exp rule and Max Weight: (a) mean delay and (b) 99^{th} %−tile delay for each user and overall system, under *low*, *medium*, and *high* symmetric loads; (c) mean and 99^{th} %−tile delay for each user and overall system under *low* symmetric traffic but when User 7 is moved to the cell edge; and (d) mean and 99^{th} %−tile delay for users and overall system for the asymmetric traffic . Each cross-tick on vertical line marks a user's performance.

rate vector $E[\boldsymbol{R}]$ and 2.35 times higher than that is stabilizable by a non-opportunistic scheduler. Fig. 4-(d) exhibits the performance results for this case.

B. Discussion of Results and Insights

Performance comparisons: As seen in Fig. 4-(a) and (b) under *low* traffic, Log and Exp rules are comparable and outperform MaxWeight. Although users see heterogeneous channels the performance they see is very similar verifying we have a good choice for the scheduling policy parameters, see Table II. However, as the traffic rate increases there are clear trends: the users' and overall means are better under the Log rule (up to 20% reduction), while the variability or spread of the 99^{th} percentile delay across users is lower under the Exp rule (the 99^{th} percentile delay spread is halved). Note, however, that all but two users have 5-70% better 99^{th} percentile delay under the Log rule versus the Exp rule. The situation is even more favorable to the Log rule at higher loads, where all users experience 20-80% lower mean and 99^{th} percentile delays versus the Exp rule (which still maintains a lower delay spread than the Log rule). Clearly for heterogenous channels, the Exp Rule's strong bias towards balancing queues is excessively compromising the realized throughput, and eventually the mean delays and tails for almost all users. Although asymptotically Exp rule should be optimal, the pre-exponent must also be playing a role in determining the systems performance.

Sensitivity: Another way to view this is that the actual performance (not the theoretical asymptotic tail) achieved by the Exp rule is more sensitive to the absolute values of $\boldsymbol{a}$. Fig. 4-(a) and (b) exhibit the degeneration in the relative performance of Exp vs Log rule for a set of fixed parameters as the load is scaled up. The RSM property of the Log rule naturally calibrates the scheduler to increased load. Similarly, comparing the *low* and the *medium* results in Fig. 4-(a) and (b) to those in Fig. 4-(c) and Fig. 4-(d), we see the performance sensitivity to changes in the channel or load characteristics. In both cases for most users the mean and 99^{th} percentile delays are better under the Log rule and in the case of heterogenous loads, i.e. Fig. 4-(d), the delay spreads are also improved. So unless parameters can be carefully tuned to possibly changing loads and unpredictable channel capacities, the Log rule appears to be a more robust scheduling policy. Intuitively, this is what one would expect from optimizing for the overall average versus worst case asymptotic tail.

Graceful degradation: Suppose the user flows correspond to buffered streaming audio sessions with a QoS requirement of 99^{th} percentile delay below 1 sec, see e.g., [17]. Under medium traffic (Fig. 4-b), all users comfortably meet the QoS requirement for both the Log and the Exp rule. However, if User 7 moves to the cell edge (Fig. 4-c), then under the Log rule, 9 out of 12 users versus 6 out of 12 for Exp rule meet the QoS requirement. If instead, the traffic loads associated with the users were to change, then as shown in Fig. 4-(d) all users meet the QoS requirement under the Log rule versus only 6 out of 12 under the Exp rule. Unless system resource is provisioned extremely conservatively, i.e. for worst case, we can expect such scenarios to arise, and this work suggests Log rule would provide a more graceful degradation of service.

VII. Conclusion

This paper has made the case not only for a new class of opportunistic scheduling policies, but also for new metrics to design and evaluate such schedulers. Our conclusion is simple, and in retrospect intuitive, a scheduler 'optimized' for the overall system performance is likely to be more robust to changes in the traffic and channel statistics than the one optimized for the worst case. The numerical results presented in this paper show that mean delay optimal schedulers exhibit radial sum-rate monotonicity (RSM). Further asymptotic results in a companion paper show that an RSM policy minimizes the exponential decay rate of the sum-queue distribution. The proposed Log rule policy is RSM and although not necessarily mean delay-optimal for a given scenario, exhibits the promised robustness vs the Exp and MaxWeight rules. The set of presented simulations (and others not included) lend support to the practical benefits of this new class of policies.

References

[1] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Comput. Networks*, vol. 41, no. 4, pp. 451–474, 2003.

[2] M. J. Neely, "Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks," *IEEE/ACM Trans. Netw.*, 2008.

[3] 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, no. 2, 2004.

[4] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The Exponential Rule," *American Mathematical Society Translations, Series 2*, vol. 207, 2002.

[5] 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.

[6] E. M. Yeh and A. S. Cohen, "Delay optimal rate allocation in multiaccess fading communications," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, 2004, pp. 140–149.

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

[8] D. Shah and D. Wischik, "Optimal scheduling algorithms for input-queued switches," *INFOCOM*, April 2006.

[9] B. Sadiq and G. de Veciana, "Optimality and large deviations of queues under the psuedo-Log rule scheduling," *Technical Report*, 2008. [Online]. Available: www.ece.utexas.edu/~gustavo/logruleoptimality.pdf

[10] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and S. Viterbi, "CDMA/HDR: A bandwidth efficient high speed wireless data service for nomadic users," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 70–77, Jul 2000.

[11] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 636–647, 2005.

[12] R. Prakash and V. V. Veeravalli, "Centralized wireless data networks with user arrivals and departures," *IEEE Trans. Inform. Th.*, vol. 53, no. 2, pp. 695–713, Feb 2007.

[13] T. Bonald, S. C. Borst, and A. Proutiére, "How mobility impacts the flow-level performance of wireless data systems," in *INFOCOM*, vol. 3, 2004, pp. 1872–1881.

[14] V. S. Borkar, "Controlled markov chains and stochastic networks," *SIAM Journal on Control and Optimization*, vol. 21, no. 4, pp. 652–666, 1983.

[15] D. P. Bertsekas, *Dynamic Programming and Optimal Control. Vol. 2*. Athena Scientific, 1995.

[16] S. Stidham and R. Weber, "A survey of Markov decision models for control of networks of queues," *Queueing Systems*, vol. 13, no. 1, pp. 291–314, Mar 1993.

[17] S. Shakkottai and A. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in HDR," in *Proceedings of the 17th International Teletraffic Congress (ITC-17)*, 2001.

Appendix

Proof of Theorem 1: We will use Foster's Criterion to show that $(\boldsymbol{Q}_k,\ k \geq 0)$ is positive recurrent for any stabilizable $\boldsymbol{\lambda}$ (see (7)). Specifically, take a (Lyapunov) function $G : \mathbb{R}_+^n \to \mathbb{R}_+$ such that $\boldsymbol{\nabla} G = \boldsymbol{g}$ and $G(\boldsymbol{0}) = 0$. Then,

$$\begin{aligned} & E^{\hat{\boldsymbol{\mu}}}[\ G(\ \boldsymbol{Q}_{k+1}\ ) - G(\ \boldsymbol{Q}_k\ )\ |\ \boldsymbol{Q}_k = \boldsymbol{q}\ ] \\ &\quad = \gamma^{-1} \sum_{i \in I} \lambda_i\ (\ G(\boldsymbol{A}_i \boldsymbol{q}) - G(\boldsymbol{q})\ )\ + \\ &\quad\quad \gamma^{-1} \sum_{i \in I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q})\ (\ G(\boldsymbol{D}_i \boldsymbol{q}) - G(\boldsymbol{q})\ )\ , \end{aligned} \tag{22}$$

where, as before, $I_{\boldsymbol{q}} = \{i :\ q_i \neq 0\}$. Since $\boldsymbol{g}$ is differentiable, define $\dot{g}_i(.) = \partial g_i(.)/\partial x_i,\ \forall i \in I$. Then G has the following Taylor expansion for any $\boldsymbol{q} \in \mathbb{Z}_+^n$,

$$\begin{aligned} G(\boldsymbol{A}_i \boldsymbol{q}) - G(\boldsymbol{q}) &= g_i(\boldsymbol{q}) + \frac{1}{2} \dot{g}_i(\boldsymbol{q} + \alpha_i \boldsymbol{e}^i) \quad \forall i \in I, \\ G(\boldsymbol{D}_i \boldsymbol{q}) - G(\boldsymbol{q}) &= -g_i(\boldsymbol{q}) + \frac{1}{2} \dot{g}_i(\boldsymbol{q} - \beta_i \boldsymbol{e}^i)\ \forall i \in I_{\boldsymbol{q}}, \end{aligned}$$

for some $\alpha_i \in [0, 1]$ and $\beta_i \in [0, 1]$ that depend on $\boldsymbol{q}$. One can rewrite (22) as follows,

$$\begin{aligned} & E^{\hat{\boldsymbol{\mu}}}[\ G(\ \boldsymbol{Q}_{k+1}\ ) - G(\ \boldsymbol{Q}_k\ )\ |\ \boldsymbol{Q}_k = \boldsymbol{q}\ ] \\ &\quad = \gamma^{-1} \sum_{i \in I} \lambda_i g_i(\boldsymbol{q})\ +\ \frac{\gamma^{-1}}{2} \sum_{i \in I} \lambda_i \dot{g}_i(\boldsymbol{q} + \alpha_i \boldsymbol{e}^i)\ - \\ &\quad\quad \gamma^{-1} \sum_{i \in I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) g_i(\boldsymbol{q})\ +\ \frac{\gamma^{-1}}{2} \sum_{i \in I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) \dot{g}_i(\boldsymbol{q} - \beta_i \boldsymbol{e}^i)\ . \end{aligned}$$

Adding and subtracting $\gamma^{-1} \sum_{i \in I \setminus I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) g_i(\boldsymbol{q})$ from the left side of above yields,

$$\begin{aligned} & E^{\hat{\boldsymbol{\mu}}}[\ G(\ \boldsymbol{Q}_{k+1}\ ) - G(\ \boldsymbol{Q}_k\ )\ |\ \boldsymbol{Q}_k = \boldsymbol{q}\ ] \\ &\quad = \gamma^{-1} \langle \boldsymbol{\lambda} - \hat{\boldsymbol{\mu}}(\boldsymbol{q}), \boldsymbol{g}(\boldsymbol{q}) \rangle\ +\ \gamma^{-1} \sum_{i \in I \setminus I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) g_i(\boldsymbol{q})\ + \\ &\quad\quad \frac{\gamma^{-1}}{2} \sum_{i \in I} \lambda_i \dot{g}_i(\boldsymbol{q} + \alpha_i \boldsymbol{e}^i)\ +\ \frac{\gamma^{-1}}{2} \sum_{i \in I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) \dot{g}_i(\boldsymbol{q} - \beta_i \boldsymbol{e}^i) \\ &\quad \leq \gamma^{-1} \langle \boldsymbol{\lambda} - \hat{\boldsymbol{\mu}}(\boldsymbol{q}), \boldsymbol{g}(\boldsymbol{q}) \rangle\ +\ \gamma^{-1} \sum_{i \in I \setminus I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) g_i(\boldsymbol{q})\ + \\ &\quad\quad \max \Big\{\ \max \left\{ \dot{g}_i(\boldsymbol{q} + \alpha_i \boldsymbol{e}^i) :\ i \in I \right\}, \\ &\quad\quad\quad \max \left\{ \dot{g}_i(\boldsymbol{q} - \beta_i \boldsymbol{e}^i) :\ i \in I_{\boldsymbol{q}} \right\} \Big\}\ . \end{aligned} \tag{23}$$

Let $\boldsymbol{u} \in \mathcal{C}$ be a service rate vector satisfying $\lambda_i < u_i$ for all $i \in I$. Define $\epsilon_1 = \gamma^{-1} \min_{i \in I} \{(u_i - \lambda_i)\}$, then clearly $\epsilon_1 > 0$. Moreover, by Lemma 1, for all $\boldsymbol{q} \in \mathbb{Z}_+^n$,

$$\langle \boldsymbol{\lambda} - \hat{\boldsymbol{\mu}}(\boldsymbol{q}), \boldsymbol{g}(\boldsymbol{q}) \rangle \leq \langle \boldsymbol{\lambda} - \boldsymbol{u}, \boldsymbol{g}(\boldsymbol{q}) \rangle \leq -\epsilon_1 \gamma |\boldsymbol{g}(\boldsymbol{q})|$$

Substituting in (23),

$$\begin{aligned} & E^{\hat{\boldsymbol{\mu}}}[\ G(\ \boldsymbol{Q}_{k+1}\ ) - G(\ \boldsymbol{Q}_k\ )\ |\ \boldsymbol{Q}_k = \boldsymbol{q}\ ] \\ &\quad \leq\quad - \epsilon_1 |\boldsymbol{g}(\boldsymbol{q})|\ +\ \gamma^{-1} \sum_{i \in I \setminus I_{\boldsymbol{q}}} \hat{\mu}_i(\boldsymbol{q}) g_i(\boldsymbol{q})\ + \\ &\quad\quad \max \Big\{\ \max \left\{ \dot{g}_i(\boldsymbol{q} + \alpha_i \boldsymbol{e}^i) :\ i \in I \right\}, \\ &\quad\quad\quad \max \left\{ \dot{g}_i(\boldsymbol{q} - \beta_i \boldsymbol{e}^i) :\ i \in I_{\boldsymbol{q}} \right\} \Big\}\ . \end{aligned} \tag{24}$$

By using (17), when $\boldsymbol{q}$ is suitably large, $g_i(\boldsymbol{q})$ for each $i \in I \setminus I_{\boldsymbol{q}}$ in the second term of the above summation can be bounded above by $\frac{\epsilon_1}{4} |\boldsymbol{g}(\boldsymbol{q})|$. Similarly, using (18), the third term of the above summation can be bounded above by $\frac{\epsilon_1}{4} |\boldsymbol{g}(\boldsymbol{q})|$. Hence, for $\boldsymbol{q}$ large enough, (24) becomes,

$$E^{\hat{\boldsymbol{\mu}}}[\ G(\ \boldsymbol{Q}_{k+1}\ ) - G(\ \boldsymbol{Q}_k\ )\ |\ \boldsymbol{Q}_k = \boldsymbol{q}\ ]\ \leq\ -\frac{\epsilon_1}{2} |\boldsymbol{g}(\boldsymbol{q})|$$

Since $0 < \epsilon < |\boldsymbol{g}(\boldsymbol{q})|$ for all large $\boldsymbol{q}$, the proof is complete.