Balancing SRPT Prioritization vs Opportunistic Gain in Wireless Systems with Flow Dynamics

Bilal Sadiq and Gustavo de Veciana
Dept. of Electrical and Computer Engineering
The University of Texas at Austin

***Abstract*—The problem of scheduling best effort flows from a dynamic population sharing a time varying wireless channel, with the objective of minimizing mean sojourn time is considered. The key tradeoff involved is between *prioritizing flows with short residual sizes* and *maximizing opportunistic capacity gain by selecting flows that currently see good channels*. This tradeoff is explicitly characterized by introducing a new queueing model that involves servers with state-dependent *capacity regions*. In the transient case and for (bounding) polymatroid capacity regions, the optimal scheduler is given and used to obtain sub-optimality bounds for various heuristics. Using a mix of analysis and simulation two regimes are described: one where fully exploiting opportunistic gain is sufficient and further using residual flow sizes for scheduling will result in only minimal reduction in mean sojourn time, and the other, where the use of this information can indeed offer significant reduction. A new scheduler is proposed which performs well in both regimes.**

I. Introduction

We consider a wireless link shared by a dynamic population of flows/users[1]. Flows of random size (bits) arrive at the base station at random times, and leave when their transfer is complete. The transmission rate supported by the wireless channel (referred to as the channel state) of each user varies randomly over time and is independent of that of other users. The scheduling problem in this context is to select which flow to serve based on the current system state (*e.g.*, residual flow sizes and channel states of the contending flows), with the objective of minimizing the mean sojourn time of flows. This scheduling problem involves a key tradeoff between *prioritizing flows with short residual sizes* and *maximizing the opportunistic gain*; below we elaborate further on this.

One *extreme* of this problem is the case where the wireless channel is in fact a constant capacity server. This reduces the system to a work-conserving G/G/1 queue, and in that case the Shortest Remaining Processing Time (SRPT) scheduler [1] is known to be optimal in a very strong sense: SRPT not only minimizes the mean sojourn time but also the number of flows in the system at all times.

However, unlike a constant capacity server, the time-varying nature of wireless channels provides a scope for opportunistic scheduling [2]–[4]. Therefore, the server's capacity is not only time-varying but also growing with the number of flows present in the system. Another *extreme* case of our problem is that where a *fixed* number of *infinitely backlogged* flows share the wireless downlink: opportunistic schedulers that maximize long-run average service rate/throughput under various fairness constraints are also well understood; see, *e.g.*, [4].

Main contributions: Using a mix of analysis and simulation, we investigate features of the above-mentioned tradeoff. The main analytical results are given in Theorems 2 and 3. Theorem 2 characterizes the competitive ratio of flow-size-oblivious opportunistic schedulers, like Proportional Fair (PF) [3] or MaxQuantile [5] [6], for a *transient* system, and shows that the presence of opportunistic gain mitigates their sub-optimality. Using this, we characterize two regimes based on the "degree" of opportunistic gain:

- A regime where the use of residual flow-size information in scheduling will *not* result in a significant reduction in flow's delay.
- A regime where optimally using flow-size information alongside channel state information *may* result in a significant reduction in flow's delay.

More specifically, but still informally, if the opportunistic capacity of the wireless channel increases rapidly in the number of users, *e.g.*, as $\log(n)$ or $\log\log(n)$ where n is the number of users, then the mean sojourn time under a purely opportunistic scheduler like MaxQuantile or PF is only about 1–20% higher than the minimum possible. If, however, the opportunistic capacity increases more *slowly* (*e.g.*, as $1 - a^n$ for $a \in [0, 1)$), a significant reduction in mean sojourn time may be achievable if schedulers exploit the residual flow-size information. Using these insights, we propose a class of schedulers which is simple to implement and offers good performance irrespective of the operating regime – this is analyzed in Theorem 3. This is part of on-going investigation.

Related work: We begin with [7] which considers a dynamic heterogenous wireless system under a *specific* flow-size oblivious scheduler, namely, Proportional Fair (PF). Using a time scale separation argument, it is shown that such a system can be modeled as a multi-class Processor Sharing (PS) queue where the server capacity/speed increases with the total number of users in the queue. This model allows one to obtain explicit formulas for the mean sojourn time as well as the queue length distribution and the stability region of the system. In this paper, we will use a similar model for the wireless channel as the above work. Note that [8] further shows that the sojourn times decrease if channels change quickly. Therefore, strictly speaking, using a time-scale separation argument to obtain a multi-class Processor Sharing model leads to optimistic performance estimates. Neither [8] nor [7] give bounds on sub-optimality of mean sojourn time under PS/PF. *These bounds are partly addressed in this paper.*

There have been various efforts to combine SRPT and opportunistic scheduling, however, few analytical results are

This research was supported in part by grants AFOSR FA9550-07-1-0428 and NSF CNS-0917067. The authors would like to thank Profs. M. Harchol-Balter and A. Wierman for the useful discussions.

[1]Each flow is associated with a unique user downloading a file from the base station; the terms flow, user, and file will sometimes be used interchangeably.

available. Most work has focused on designing and investigating heuristics through a mix of simulation and analysis.

Both [9] and [10] consider a *transient* system, *i.e.*, the problem of transmitting a fixed set of finite sized files from the base station to their respective destination users, with the objective of minimizing the total expected sojourn time. [9] shows that the problem can be set up as a Markov decision process (MDP) with a finite state space and numerically solved. However, perhaps due to having a very large state-space for any useful system, it is difficult to characterize any structural properties of the optimal scheduler from the computed solution. [10] explores various ways of combining SRPT and opportunistic scheduling and provides bounds on mean sojourn time for the proposed heuristics. *In this paper, we will show that the transient system of [9] and [10] can be modeled as a deterministic dynamic program, and explicitly give the optimal scheduler and sojourn times for certain insightful examples.*

Both [11] and [12] consider dynamic systems, however, [11] considers only such wireless systems where the *current* channel state information is not available to the scheduler. [12] motivates various heuristics that combine SRPT and opportunistic scheduling, and compares them via extensive simulation. From the presented results, one can make the qualitative observation that all the schedulers which perform well in various situations do *not* compromise opportunistic gain or do so only minimally. [12] also points out the necessary condition for the stabilizability of a dynamic system and suggests that this condition may also be *sufficient*, but stops short of a formal proof.

Recently, [13] has formally characterized the stability region of such a system, and (implicitly) shown that, asymptotically in the number of flows, any throughput-optimal scheduler must fully exploit the opportunistic gain. It is also shown that a version of MaxWeight scheduler which prioritizes flows with *long* residual sizes can indeed be unstable. [14] shows the version of MaxWeight which prioritizes flows with long *current sojourn times* is nevertheless throughput optimal, however, perhaps not suitable for use in a real wireless system: the scheduler may excessively compromise opportunistic capacity until a significant fraction of the contending flows has large and nearly equal sojourn times. Moreover, the relation between the *residual flow-size* and its *current delay* is unclear.

[15] deals with designing practically viable throughput optimal schedulers and proposes a measurement-based MaxQuantile-like scheduler. The problem of scheduling a mix of best effort and QoS flows is also considered. Once again, the schedulers which seem to out perform in various simulation settings are those which do not or only minimally compromise the opportunistic gain.

II. System Model

We will first define a *heterogeneous* wireless system where the heterogeneity in the users' wireless channels is captured by a single parameter, namely, their mean supported transmission rate. In the next subsection, we will translate the heterogeneous system into an equivalent system with *homogeneous* wireless channels. The latter system permits certain simplifications for subsequent exposition and analysis. The details are as follows.

A. Wireless system with heterogeneous channels

We consider a continuous time system. Let $\mathcal{X} \subseteq \mathbb{R}^2$ denote the connected coverage area of a base station. New users arrive at random times and random locations in $\mathcal{X}$, request to download (or upload) a file of random size, and leave once the download completes. The details are as follows.

Using an index set $\mathcal{I} = \{1, 2, \cdots\}$, we uniquely index each user that enters the system. For each user $i \in \mathcal{I}$, let the random variables $A_i \in \mathbb{R}$, $\boldsymbol{X}_i \in \mathcal{X}$, and $\tilde{B}_i \in \mathbb{R}_+$ denote its arrival time, location, and the size of the file requested, respectively. Users are indexed in order of their arrival times, *i.e.*, for $i < j$, we have $A_i \leq A_j$. We assume that,

- $(A_i,\ i \in \mathcal{I})$ correspond to the jump times of a homogenous Poisson process of intensity λ;
- $\boldsymbol{X}_i,\ i \in \mathcal{I}$ are i.i.d. with density $f_{\boldsymbol{X}}(\cdot)$, *i.e.*, for any measurable $\mathcal{B} \subseteq \mathcal{X}$, we have $\mathbb{P}(\boldsymbol{X}_1 \in \mathcal{B}) = \int_{\mathcal{B}} f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}$;
- $\tilde{B}_i,\ i \in \mathcal{I}$ are i.i.d. with density $f_{\tilde{B}}(\cdot)$.

The time-varying channel state of the i^{th} user is given by a stationary random process $\left(\tilde{R}_i(t),\ t \in \mathbb{R}\right)$, where $\tilde{R}_i(t) \in \mathbb{R}_+$ denotes the data/service rate supported by the user's channel at time t. Note that even though $\tilde{R}_i(t)$ is defined for all t, it is relevant only over the time interval that user i spends in the system.

Assumption 1: We assume the following regrading the distribution of $\left(\tilde{R}_i(t),\ t \in \mathbb{R}\right)$.[2]

(i) Marginal distribution: Conditional on $\boldsymbol{X}_i = \boldsymbol{x}$, we assume that $\left(\tilde{R}_i(t),\ t \in \mathbb{R}\right)$ has the same marginal distribution as that of $m(\boldsymbol{x})R$, where R is a unit mean random variable and $m : \mathcal{X} \to [\xi_1, \xi_2] \subset (0, \infty)$ is a given continuous function that captures the impact of pathloss/shadowing on the achievable transmission rate from the base station to location $\boldsymbol{x} \in \mathcal{X}$.

(ii) Separation of time-scales: We assume that the channels change on a much faster time-scale than that of the user dynamics (arrivals and departures). So, for example, the channel coherence time – the smallest τ such that $\tilde{R}_i(t)$ and $\tilde{R}_i(t+\tau)$ are independent – is much smaller than the mean inter-arrival time and mean sojourn time.

Let random set $Q(t) \subset \mathcal{I}$ denote the set of users with on going file transfers at time t, *i.e.*, the users which have arrived but have not completed their file download by time t. We will refer to the users in $Q(t)$ as the users *present* at time t, and to $Q(t)$ as the queue at time t. Cardinality of the set $Q(t)$, *i.e.*, the number of users present in the system at time t, is denoted by $|Q(t)|$. For any $i \in Q(t)$, let $L_i(t) > 0$ denote the i^{th} user's residual file size.

We will follow the convention that uppercase letters, *e.g.* $Q(\cdot)$, denote random quantities, whereas, lowercase letters, *e.g.* $q(\cdot)$, denote particular realizations. Bold letters are

[2] Assumption 1*(i)* will be used in the next subsection to *shift* the heterogeneity in the channels of users to their arrival processes, in the same manner as done in [7]. Assumption 1*(ii)* will be used in Section IV-B to remove the randomness due to time-varying channels *without* losing any opportunistic gain associated with such channels. This is perhaps a key assumption which enables most of the results presented in this chapter and in [7].

reserved for vectors. Also, we will make the natural distinction between "increasing" and "strictly increasing", and so forth.

The state of the system at time t is given by,

$$S_t \equiv \Big(\, Q(t); \; \big(L_i(t), \; i \in Q(t)\big); \; \big(\tilde{R}_i(t), \, i \in Q(t)\big); \\ \big((\boldsymbol{X}_i, A_i), \, i \in Q(t)\big) \, \Big) \; , \quad (1)$$

We assume that the current system state is available for making a scheduling decision, and define a *scheduler* or a *scheduling policy* as a mapping from the current system state to the current queue. That is, when the system is in state s_t, the scheduler $i^*(\cdot)$ selects a user $i^*(s_t) \in q(t)$ to receive service. If for a given sample path we have $i \in q(\cdot)$ over some interval $[t_1, t_2]$, then the total service received by user i over $[t_1, t_2]$ is given by $\int_{t_1}^{t_2} \tilde{r}_i(t) \mathbb{1}\{i^*(s_t) = i\} dt$.

B. Equivalent system with i.i.d. channels but heterogeneous file sizes

In the system described above, the channel state process of a user at location $\boldsymbol{x}$ has the same marginal distribution as that of $m(\boldsymbol{x})R$, and therefore, a mean supported transmission/service rate of $m(\boldsymbol{x})$. Consider a *second* system (constructed on the same probability space) where all users' channels have the same marginal distribution as that of R, but the file sizes of users at location $\boldsymbol{x}$ are scaled by a factor of $1/m(\boldsymbol{x})$. At any time t, the system state of the second system can be derived from the system state of the first system by appropriately scaling $\big(L_i(t), \; i \in Q(t)\big)$ and $\big(\tilde{R}_i(t), \; i \in Q(t)\big)$ based on $\big(\boldsymbol{X}_i, \; i \in Q(t)\big)$, and keeping the remaining elements of S_t, in particular, $Q(t)$, unchanged. That is, any scheduler for the first system can be translated into a scheduler for the second system, such that the user arrival and departure times, as captured by $Q(t)$, are identical in the two systems. The second system however has the *advantage* that the channel states of the users are independent of their identities and locations.

Therefore, from this point on, we exclusively focus on the *second* system where the user arrivals $(A_i, i \in \mathcal{I})$ and locations $(X_i, i \in \mathcal{I})$ are statistically identical to those in the first system, but the user channels $\big((R_i(t), t \in \mathbb{R}), \; i \in \mathcal{I}\big)$ and file sizes $(B_i, i \in \mathcal{I})$ have the following modified marginal distributions: for all $i \in \mathcal{I}$,

- the channel process $\big(R_i(t), \; t \geq 0\big)$ has the same marginal distribution as that of the random variable R,
- the file size B_i has density $f_B(\cdot)$ given by,

$$f_B(b) \equiv \int_{\mathcal{X}} m(\boldsymbol{x}) f_{\tilde{B}}\big(m(\boldsymbol{x})b\big) f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}, \quad b \in \mathbb{R}_+.$$

We continue to model a scheduler $i^*(\cdot)$ as a mapping from the current system state to the current queue, $i^*(s_t) \in q(t)$.

Next we define a new queueing model using a server with a state-dependent but deterministic *capacity region* – such a server will subsequently be used to replace the randomly-varying channels in the above system *without* losing the opportunistic capacity gains associated with such channels.

III. Servers with state-dependent capacity regions and $M/GI/\boldsymbol{\mathcal{C}}$ queue

Let $\mathcal{C}_1, \mathcal{C}_2, \ldots$ be sequence of *nested* capacity regions with the following properties:

(i) For any $n \geq 1$, $\mathcal{C}_n$ is a compact, convex, and coordinate-convex region of $\mathbb{R}_+^n$.

(ii) $\mathcal{C}_n$ is symmetric, *i.e.*, if a rate vector $\boldsymbol{\mu} \equiv (\mu_1, \cdots, \mu_n) \in \mathcal{C}_n$, then any vector given by a permutation of $\boldsymbol{\mu}$'s components also lies in $\mathcal{C}_n$.

(iii) For k and n such that $1 \leq k \leq n$, we have $\mathcal{C}_k = \mathbb{R}_+^k \cap \mathcal{C}_n$.

Let $\boldsymbol{\mathcal{C}} \equiv (\mathcal{C}_n, \; n \geq 1)$. Consider a server with the following property: if there are n jobs (files) in the system, the server can process them simultaneously according to any speed (rate) vector from the region $\mathcal{C}_n$. For example, suppose over some interval $[t_1, t_2]$ the n files are served at rate $\boldsymbol{\mu}(t) \in \mathcal{C}_n$, then the file backlog $\boldsymbol{l}(t) \equiv (l_i(t), \; 1 \leq i \leq n)$ over $t \in [t_1, t_2]$ evolves as,

$$\boldsymbol{l}(t) = \left(\boldsymbol{l}(t_1) - \int_{t_1}^{t} \boldsymbol{\mu}(\tau) d\tau \right)^+ .$$

By replacing the unit capacity server in a conventional $M/GI/1$ queue with the server described above, we obtain a new queueing model which we refer to as an $M/GI/\boldsymbol{\mathcal{C}}$ queue. Note that the queues $M/GI/1$ and $M/GI/\infty$ queues are special cases of the $M/GI/\boldsymbol{\mathcal{C}}$ queue. Indeed the latter reduces to $M/GI/1$ if $\mathcal{C}_n$ is unit n-simplex, *i.e.*,

$$\mathcal{C}_n = \left\{ \boldsymbol{\mu} \in \mathbb{R}_+^n : \; \|\boldsymbol{\mu}\|_1 \leq 1 \right\}, \quad n = 1, 2, \ldots$$

and to $M/GI/\infty$ if $\mathcal{C}_n$ is unit n-cube, *i.e.*,

$$\mathcal{C}_n = \left\{ \boldsymbol{\mu} \in \mathbb{R}_+^n : \; \|\boldsymbol{\mu}\|_\infty \leq 1 \right\}, \quad n = 1, 2, \ldots \; .$$

As will become clear in the next section, we will be interested in server capacity regions that are *in between* unit simplices and unit cubes.

Next, we will show that the system with time-varying channels and flow dynamics, described in Section II-B, can be reduced to an instance of an $M/GI/\boldsymbol{\mathcal{C}}$ queue by *appropriately defining the capacity regions* $\mathcal{C}_n, \; n \geq 1$ (Section IV-A), and *using a time-scale separation argument* (Section IV-B).

IV. Dynamic wireless system as an $M/GI/\boldsymbol{\mathcal{C}}$ queue

We will set $\mathcal{C}_n$ equal to the opportunistic capacity region of a *static* wireless system, *i.e.*, a system with a fixed number n of infinitely-backlogged flows. A formal description of these regions follows.

A. Opportunistic capacity region

Throughout this subsection, we will assume a static system with a fixed number n of users. We define the opportunistic capacity region $\mathcal{C}_n \subset [0,1]^n$ of such a system as the set of longrun average service rates that are jointly achievable by the n users. The specific definition is as follows.

For any $\boldsymbol{r} = (r_i, \; 1 \leq i \leq n) \in \mathbb{R}_+^n$, let $\mathcal{C}(\boldsymbol{r}) \subset \mathbb{R}_+^n$ be the convex hull of n points,

$$(r_1, 0, \cdots, 0), \; (0, r_2, 0, \cdots, 0), \; \ldots, \; (0, \cdots, 0, r_n).$$

The opportunistic capacity region of n channels, $\mathcal{C}_n$, is defined as,

$$\mathcal{C}_n \equiv \big\{ \boldsymbol{\mu} \in [0,1]^n : \boldsymbol{\mu} \leq \mathbb{E}\left[\boldsymbol{v}(\boldsymbol{R})\right] \text{ for some } \boldsymbol{v}(\cdot) \text{ that satisfies} \\ \boldsymbol{v}(\boldsymbol{r}) \in \mathcal{C}(\boldsymbol{r}) \text{ for all } \boldsymbol{r} \in \mathbb{R}_+^n \big\},$$

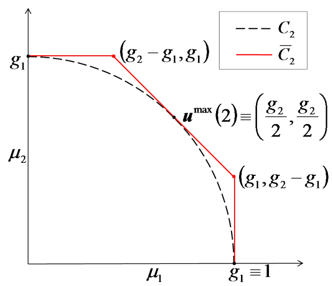

Fig. 1. Capacity region $\mathcal{C}_2$ and the tightest polymatroid upper bound $\bar{\mathcal{C}}_2$.

where expectation is with respect to random $\boldsymbol{R}$ drawn from the n-product of the distribution of R.

Any extremal point of the capacity region $\mathcal{C}_n$ is achievable by a very simple, distribution-oblivious, opportunistic scheduler; for details see [4] or [7, Lemma 2.2].

For later use, in the following we define a polymatroid outer bound for capacity region $\mathcal{C}_n$.

The tightest polymatroid region containing $\mathcal{C}_n$: Let $R_1, R_2, \ldots$ be i.i.d. copies of R. For any integer $k > 0$, let

$$g_k \equiv \mathbb{E}\left[\ \max(R_1, \cdots, R_k)\ \right], \tag{2}$$

and let $g_0 \equiv 0$. This function has the following interpretation: no single user can get an average rate exceeding g_1, no two users can get a total average rate exceeding g_2, and so on. We will refer to $(g_k,\ k \geq 0)$ as the opportunistic *capacity function*. We have that g_k is concave increasing in k, *i.e.*, for any $k \geq 0$, $g_{k+1} - g_k \geq g_{k+2} - g_{k+1} \geq 0$, and therefore, the capacity per user g_k/k is decreasing in k.

Then using $(g_k,\ k \leq n)$, we can define a polymatroid $\bar{\mathcal{C}}_n$ as follows,

$$\bar{\mathcal{C}}_n \equiv \left\{ \boldsymbol{\mu} \in [0,1]^n :\ \forall \mathcal{K} \subseteq \{1, \cdots, n\},\ \sum_{k \in \mathcal{K}} \mu_k \leq g_{|\mathcal{K}|} \right\}.$$

See Fig. 1 for an illustration of $\mathcal{C}_n$ and $\bar{\mathcal{C}}_n$. For any n, we have that $\mathcal{C}_n \subseteq \bar{\mathcal{C}}_n$. We also note the following.

- In general, we cannot completely construct $\mathcal{C}_n$ from $(g_k, k \leq n)$ alone. Nevertheless, we can completely construct the outer bound $\bar{\mathcal{C}}_n$.
- Region $\bar{\mathcal{C}}_n$ is the tightest polymatroid outer bound for $\mathcal{C}_n$.

In special cases of time-varying channels, *e.g.*, on-off channels or information theoretic block fading multiaccess channels (as used in [16]), we have $\mathcal{C}_n = \bar{\mathcal{C}}_n$. In fact, an alternative way to define the outer bound $\bar{\mathcal{C}}_n$ is as follows: $\bar{\mathcal{C}}_n$ is the region obtained by replacing $\mathcal{C}_n(\boldsymbol{r})$ in the definition of $\mathcal{C}_n$ by the tightest polymatroid containing $\mathcal{C}_n(\boldsymbol{r})$. Let $\bar{\boldsymbol{\mathcal{C}}} \equiv \left(\bar{\mathcal{C}}_n,\ n \geq 1\right)$.

Remark 1: We say a vector $\boldsymbol{a} \in \mathbb{R}^n$ dominates a vector $\boldsymbol{b} \in \mathbb{R}^n$ if $\boldsymbol{a} \geq \boldsymbol{b}$, and a point is maximal in a region if it is not dominated by any other point of that region. Note that a polymatroid capacity region has the *nice* property that all maximal elements have the same L_1 norm (the total service rate). For example, the L_1 norm of all maximal elements of $\bar{\mathcal{C}}_n$ is g_n. Therefore, a scheduler for an $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ queue will *not* have to tradeoff *maximizing total service rate* with *prioritizing short files*.

Let $\boldsymbol{u}^{max}(n) \equiv \left(\frac{g_n}{n}, \cdots, \frac{g_n}{n}\right) \in \mathbb{R}_n$. We have that $\boldsymbol{u}^{max}(n)$ lies in $\mathcal{C}_n$, and $g_n = \|\boldsymbol{u}^{max}(n)\|_1 = \max_{\boldsymbol{\mu} \in \mathcal{C}_n} \|\boldsymbol{\mu}\|_1 = \max_{\boldsymbol{\mu} \in \bar{\mathcal{C}}_n} \|\boldsymbol{\mu}\|_1$. Therefore, we will refer to $\boldsymbol{u}^{max}(n)$ as the *max-sum-rate point* of region $\mathcal{C}_n$ as well as $\bar{\mathcal{C}}_n$.

B. Time-scale separation argument and reduction to an $M/GI/\boldsymbol{\mathcal{C}}$ queue

We return to the homogeneous dynamic system introduced in Section II-B, and recall the Assumption 1*(ii)* that channels vary at a much faster time-scale than that of the user dynamics. Therefore, we can further assume that at any time t, conditional on $|Q(t)| = n$, the n users can be jointly served at any rate from the capacity region $\mathcal{C}_n$. That is, we can redefine the scheduler as a mapping from

$$\big(q(t);\ \big(l_i(t),\ i \in q(t)\big);\ \big((\boldsymbol{x}_i, a_i),\ i \in q(t)\big)\big)$$

to the capacity region $\mathcal{C}_{|q(t)|}$. For example, if for $t \in [t_1, t_2]$, we have $|q(t)| = n$ and the scheduler serves at rate $\boldsymbol{\mu}(t) \in \mathcal{C}_n$, then the file backlog $\boldsymbol{l}(t) \equiv (l_i(t),\ i \in q(t_1))$ over $t \in [t_1, t_2]$ evolves as $\boldsymbol{l}(t) = \boldsymbol{l}(t_1) - \int_{t_1}^{t} \boldsymbol{\mu}(\tau) d\tau$.

This new system is simply an $M/GI/\boldsymbol{\mathcal{C}}$ queue, where the arrivals are Poisson with rate λ, the file sizes have density $f_B(\cdot)$, and the server capacity region at time t depends on only the element $|q(t)|$ of the system state. The reason for converting the original system with heterogeneous channels to the one with homogeneous channels now becomes clear: it removed the dependence of the server capacity region $\mathcal{C}_{|q(t)|}$ on the *locations* of the users in $q(t)$.

The work in [7] can be seen as analyzing an $M/GI/\boldsymbol{\mathcal{C}}$ queue under the scheduler that, conditional on $|Q(t)| = n$, serves at the max-sum-rate point $\boldsymbol{u}^{max}(n) \in \mathcal{C}_n$, and therefore, simultaneously serves each of the n users in the current queue at rate g_n/n. Following [7], we will call this the Opportunistic Processor Sharing (OPS) scheduler. This scheduler is clearly oblivious to file sizes.

Remark 2: A few observations regarding the usefulness of the $M/GI/\boldsymbol{\mathcal{C}}$ queueing model are in order. By reducing the original system to an $M/GI/\boldsymbol{\mathcal{C}}$ queue, we have replaced the state-dependent and randomly varying server (*i.e.*, wireless channel) with a state-dependent but deterministic server *without* losing the opportunistic gain associated with a randomly varying server. In the latter system,

- the tradeoff mentioned in the beginning of Section **??** is manifested as a tradeoff between *maximizing total service rate* (*e.g.*, by picking the rate point $\boldsymbol{u}^{max}(\cdot)$) and *giving more rate to shorter files*;
- moreover, the *extent* of opportunistic gain present in the channel is explicitly characterized by the marginal rate of increase of opportunistic capacity function $\left(g_n,\ n \geq 0\right)$.

Therefore, the $M/GI/\boldsymbol{\mathcal{C}}$ queue is better suited to investigate the scheduling problem, even if by simulation methods only.

The $M/GI/\boldsymbol{\mathcal{C}}$ queue also seems more amenable to an analytical solution than the original system; indeed, [7] gives analytical results for at least one particular scheduler for this queue.

We also note that the $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ queue – a *simpler* to analyze than the $M/GI/\boldsymbol{\mathcal{C}}$ queue due to the absence of above-mentioned tradeoff – can be used to obtain a lower bound on the optimal mean sojourn time in $M/GI/\mathcal{C}$ queue. To this end, in the next section, we will characterize the optimal scheduler under polymatroid capacity regions $(\bar{\mathcal{C}}_k,\ k>0)$ for a *transient* system, and use this to bound the sub-optimality of OPS.

V. Transient system

Throughout this section, we consider a transient system that starts with a given number n of files, *i.e.*, $q(0)=\{1,\cdots,n\}$, of arbitrary sizes $\boldsymbol{l}(0)\equiv(l_1(0),\cdots,l_n(0))\in(0,\infty)^n$, and there are no further arrivals. The opportunistic capacity regions $\mathcal{C}$ and $\bar{\mathcal{C}}$, along with the corresponding concave increasing capacity function $(g_k,\ k\geq 0)$ with $g_0=0$, are specified. Since there are no further arrivals, the system state at time t is simply given by $\big(q(t);\big(l_i(t),\ i\in q(t)\big)\big)$. When there are $k\leq n$ files in the system, they can be served at any rate from the region $\mathcal{C}_k\subseteq\mathcal{C}_n$ (or $\bar{\mathcal{C}}_k\subseteq\bar{\mathcal{C}}_n$, depending upon the context). Let Ψ and $\bar{\Psi}$ be the set of all functions (schedulers) that map any system state $\big(q(t);\ \boldsymbol{l}(t)\big)$ to a rate vector in $\mathcal{C}_{|q(t)|}$ and $\bar{\mathcal{C}}_{|q(t)|}$, respectively.

For any scheduler $\boldsymbol{\psi}(\cdot)$ and initial state $\big(q(0);\ \boldsymbol{l}(0)\big)$, let $\Big(\big(q^{\psi}(t),\boldsymbol{l}^{\psi}(t)\big),\ t\geq 0\Big)$ be the associated system sample path. The total sojourn time (or cost) under $\boldsymbol{\psi}(\cdot)$ is denoted by $c_{\psi}(\boldsymbol{l}(0);|q(0)|)$ and given as follows,

$$c_{\psi}(\boldsymbol{l}(0);|q(0)|)\equiv\int_0^{\infty}|q^{\psi}(t)|dt. \tag{3}$$

We are interested in schedulers in Ψ that minimize the cost starting in any state $\big(q(0);\boldsymbol{l}(0)\big)$. We naturally have $\Psi\subseteq\bar{\Psi}$, and therefore,

$$\min_{\bar{\boldsymbol{\psi}}\in\bar{\Psi}}c_{\bar{\psi}}(\cdot)\leq\min_{\boldsymbol{\psi}\in\Psi}c_{\psi}(\cdot). \tag{4}$$

However, only schedulers from Ψ are feasible in an actual system which has a server with capacity region $\mathcal{C}_n$. The rest of this section is organized as follows:

- An optimal scheduler in $\bar{\Psi}$ along with an expression for optimal cost is given in Section V-A.
- An expression for cost under OPS, which lies in Ψ (and thus also in $\bar{\Psi}$), is given in Section V-B.
- The competitive ratio of OPS under particular capacity functions is obtained in Section V-C;
- That the performance under OPS deteriorates as n becomes large and opportunistic capacity function g_n saturates is elaborated in Section V-D.
- A scheduler belonging to the set Ψ but with cost *close* to the that of the optimal scheduler in $\bar{\Psi}$ is proposed in Section V-E.

All proofs are omitted and relegated to [17].

A. Optimal scheduler for polymatroid capacity regions $\bar{\mathcal{C}}$

We begin by defining the *Shortest Remaining Processing Time - Highest Possible Rate* (SRPT-HPR) scheduler for polymatroid capacity regions, *i.e.*, SRPT-HPR lies in $\bar{\Psi}$.

Definition 1: Consider a system with a concave increasing opportunistic capacity function $(g_k,k\geq 0)$ and the associated capacity regions $\big(\bar{\mathcal{C}}_k,k\geq 0\big)$. At any time t and for any integer $n\equiv|q(t)|>0$, let $l_1(t)\leq l_2(t)\leq\ldots\leq l_n(t)$ be the remaining file sizes of the n files. Then SRPT-HPR serves the files at rate vector $\boldsymbol{s}\big(\boldsymbol{l}(t);n\big)=\big(s_i(\boldsymbol{l}(t);n),\ 1\leq i\leq n\big)\in\bar{\mathcal{C}}_n$ given as follows: for any $i\in\{1,\cdots,n\}$, we have,

$$s_i\big(\boldsymbol{l}(t);n\big)\equiv g_i-g_{i-1}\ .$$

Note that service rate allocation under SRPT-HPR depends only on the ordering of file sizes, and the shortest file is given as much rate as possible, *i.e.*, g_1, and having made that allocation, the next shortest file is given as much rate as possible, *i.e.*, g_2-g_1, and so on.

Note that a server with a polymatroid capacity region $\bar{\mathcal{C}}_n$ can be decomposed into n conventional servers (like the one in queue $M/GI/1$) with different speeds, such servers are usually called parallel *uniform* servers [18]. More specifically, consider n servers where the speed of the k^{th} server is given by g_k-g_{k-1}. Then each maximal vertex of the polymatroid $\bar{\mathcal{C}}_n$ corresponds to an allocation of the n servers to the n users (with one user per server and one server per user), and vice versa. All other maximal points of $\bar{\mathcal{C}}_n$ can further be achieved by preempting and time-sharing the n servers among the users.

Therefore, the problem of minimizing the total sojourn time under a polymatroid capacity region $\bar{\mathcal{C}}_n$ is the same as minimizing the total sojourn time under n parallel uniform servers with preemption permitted. For the latter system, [19] (see [18], pp. 134–136 for a proof) showed that the optimal scheduler is the one which allocates the fastest available server to the shortest available job/file – this indeed translates to SRPT-HPR in our setting. An alternative proof of the optimality of SRPT-HPR with an explicit expression for total sojourn time is given in [17]. We summarize the results in the following lemma and theorem.

Lemma 1: For a given concave increasing capacity function $(g_k,\ k\geq 0)$, and initial state $\big(q(0);(l_k(0),\ 1\leq k\leq|q(0)|)\big)$, the cost under SRPT-HPR $c_s(\cdot)$ is given by,

$$c_s\big(\boldsymbol{l}(0);|q(0)|\big)=\sum_{k=1}^{|q(0)|}\theta_k l_{(k)}(0)\ , \tag{5}$$

where $l_{(k)}(0)$ denote the k^{th} largest file, *i.e.*, $l_{(1)}(0)\geq\cdots\geq l_{(n)}(0)$, and $(\theta_k,\ k>0)$ are given as follows. Let $\theta_0\equiv 0$ and for $k\geq 0$, let $\Delta_k\equiv g_{k+1}-g_k$. Then for all $n\geq 0$,

$$(\boldsymbol{\theta}*\boldsymbol{\Delta})(n)\equiv\sum_{k=0}^{n}\Delta_k\theta_{n-k}=n. \tag{6}$$

Using (6), θ_{n+1} can be computed from $(\theta_k,0\leq k\leq n)$ for any given capacity function $(g_k,k\geq 0)$.

Remark 3: θ_k can be interpreted as *time* (or *cost*) *per unit data in the* k^{th} *file*. As shown in [17], θ_k is increasing in k.

Theorem 1: Consider a transient system with polymatroid capacity regions $\big(\bar{\mathcal{C}}_k,k\geq 0\big)$ and starting with any number $|q(0)|>0$ of files of arbitrary sizes $\boldsymbol{l}(0)=(l_k(0),\ k\in q(0))$. Then SRPT-HPR minimizes the cost given by (3), *i.e.*,

$$c_s\big(\boldsymbol{l}(0);|q(0)|\big)=\min_{\boldsymbol{\psi}\in\bar{\Psi}}c_{\psi}\big(\boldsymbol{l}(0);|q(0)|\big).$$

Next, we define OPS scheduler which is not limited to polymatroid capacity regions.

B. OPS scheduler for capacity regions $\bar{\mathcal{C}}$ and $\mathcal{C}$

The scheduling decision under OPS depends only on $(g_k, k \geq 0)$ and is the same for $\mathcal{C}$ and $\bar{\mathcal{C}}$. It is defined as follows.

Definition 2: Consider a system with a concave increasing opportunistic capacity function $(g_k, k \geq 0)$ and the associated capacity regions $(\mathcal{C}_k, k \geq 0)$ or $\left(\bar{\mathcal{C}}_k, k \geq 0\right)$. At any time t and for any integer $n \equiv |q(t)| > 0$, OPS serves each of the n files at rate $\frac{g_n}{n}$. Therefore, the scheduling decision for capacity region $\mathcal{C}_n$ is identical to that for $\bar{\mathcal{C}}_n$.

Lemma 2: For a given capacity function $(g_k,\ k \geq 0)$, and initial state $\left(q(0); (l_k(0),\ 1 \leq k \leq |q(0)|)\right)$, the cost under OPS $c_p(\cdot)$ is given by,

$$c_p\left(\boldsymbol{l}(0); |q(0)|\right) = \sum_{k=1}^{|q(0)|} \pi_k l_{(k)}(0)\ , \qquad (7)$$

where, for any $k > 0$,

$$\pi_k \equiv \frac{k^2}{g_k} - \frac{(k-1)^2}{g_{k-1}}. \qquad (8)$$

C. Competitive ratio of OPS

For a given transient system – *i.e.*, a capacity function $(g_k, k \geq 0)$ along with the *actual* capacity regions $\mathcal{C}$ and the *outer bound* $\bar{\mathcal{C}}$ – we will refer to the ratio $\frac{c_p(\cdot)}{c_s(\cdot)}$ as the competitive ratio of OPS when starting in state $(\cdot)$. When the starting state or an element of it is not specified, the competitive ratio is intended to mean the supremum over unspecified elements.

Remark 4: Competitive ratio has the following interpretation. Consider a transient system starting with n files of sizes $\boldsymbol{l} = (l_1, \cdots, l_n)$. Given only $(g_k, 0 \leq k \leq n)$, the largest possible capacity region that the server can have is the polymatroid $\bar{\mathcal{C}}_n$. Therefore, the smallest possible cost a scheduler can have is $c_s(n; \boldsymbol{l})$. Then, the competitive ratio $\frac{c_p(n;\boldsymbol{l})}{c_s(n;\boldsymbol{l})}$ gives us a bound on the sub-optimality of OPS when starting in state $(\{1, \cdots, n\}; \boldsymbol{l})$. Moreover, this bound is the tightest possible if only $(g_k, 0 \leq k \leq n)$ is specified instead of the complete actual capacity region $\mathcal{C}_n$.

It is easy to show that if $g_k = k$ for all $k \geq 0$ (regions $\mathcal{C}_k$ and $\bar{\mathcal{C}}_k$ are k-cubes), then $\theta_k = \pi_k = 1$ and we simply have $\frac{c_p(\cdot)}{c_s(\cdot)} = 1$. In the following, we investigate this ratio for capacity functions of the type $g_k = \frac{1-a^k}{1-a}$ (equivalently, $\Delta_k = a^k$) for any fixed $a \in [0, 1)$. Prior to doing so, we offer three comments on this choice of capacity function.

First, it corresponds to the capacity function of a system where R is an on-off random variable, with $\mathbb{P}\left(R = 0\right) = a$ and $\mathbb{P}\left(R = \frac{1}{1-a}\right) = 1 - a$.

Second, parameter a provides a measure for the opportunistic gain present in the system: for any fixed k, the higher the value of a, the greater the marginal opportunistic capacity Δ_k. Setting $a = 0$ corresponds to a non-opportunistic work-conserving system with a single unit speed server.

Third, if R takes values in a bounded set $[0, r^{max}]$, then R is stochastically dominated by an on-off random variable $\tilde{R} \in \{0, r^{max}\}$ with $\mathbb{P}\left(\tilde{R} = r^{max}\right) \equiv \mathbb{E}[R]/r^{max}$. Therefore, the capacity function associated with R can be bounded above by $(g_k, k \geq 0)$ with $a = 1 - \mathbb{E}[R]/r^{max}$. The significance of the results presented below is either a monotonicity in a or independence from a, therefore, this simple choice of capacity function still provides significant insight.

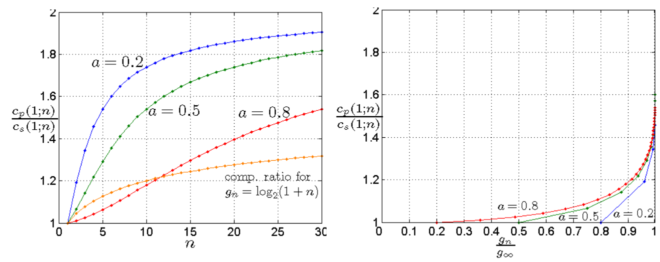

Fig. 2. Competitive ratio of OPS plotted versus (left) n, (right) $\frac{g_n}{g_\infty}$.

Fact 1: With $g_k = \frac{1-a^k}{1-a}$ for some fixed $a \in [0, 1)$, we can take the z-transform of both sides of (6) and show that the corresponding $(\theta_k,\ k > 0)$ are given by,

$$\theta_k = \frac{k}{g_\infty} + a, \qquad (9)$$

where $g_\infty \equiv \frac{1}{1-a}$.

To indicate the dependence of cost on the parameter a, in this subsection we will explicitly write $c_s(\cdot;\cdot;a)$ and $c_p(\cdot;\cdot;a)$. Moreover, $\mathbf{1}$ in $c(\mathbf{1}; n; \cdot)$ will denote an n dimensional vector of all ones. We have the following result; see [17] for proof.

Theorem 2: For any $a \in [0, 1)$, let $g_k = \frac{1-a^k}{1-a}$ for all $k \geq 0$. Let $c_s(\boldsymbol{l}; n; a)$ and $c_p(\boldsymbol{l}; n; a)$ be the costs under SRPT-HPR and OPS respectively, for capacity function parameter a and when starting with n files of sizes given in $\boldsymbol{l} \equiv (l_1, \ldots, l_n)$. Then for any fixed a,

$$\sup_{n\geq 1, \boldsymbol{l}\in(0,\infty)^n} \frac{c_p(\boldsymbol{l}; n; a)}{c_s(\boldsymbol{l}; n; a)} = \sup_{n\geq 1} \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = 2. \qquad (10)$$

More precisely, as $n \to \infty$, we have,

$$\frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} \uparrow 2.$$

Also, for any fixed $n > 0$,

$$\sup_{a\in[0,1), \boldsymbol{l}\in(0,\infty)^n} \frac{c_p(\boldsymbol{l}; n; a)}{c_s(\boldsymbol{l}; n; a)} = \sup_{a\in[0,1)} \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = \frac{2}{1 + 1/n}.$$

More precisely, as $a \to 0$, we have,

$$\frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} \uparrow \frac{2}{1 + 1/n}.$$

D. Discussion

It follows from Theorem 2 that for any fixed n, the competitive ratio of OPS *decreases* as the parameter a – a measure of opportunistic gain – increases; see Fig. 2 (left).

However, *irrespective of* a, the competitive ratio of OPS is exactly 2, which is achieved only as a monotonic limit in n. An intuitive explanation for this is as follows: the opportunistic capacity g_k is bounded above by the limit $g_\infty \equiv \frac{1}{1-a}$, therefore for any fixed a and correspondingly large n (where n is the number of files), the opportunistic system *appears* non-opportunistic with constant capacity g_∞; thus the competitive ratio 2 which is the same as in the case of non-opportunistic systems ($a = 0$).

See Fig. 2 (right) for a plot of competitive ratio as a function of n versus the *capacity ratio* $\frac{g_n}{g_\infty}$. We note that the competitive ratio is under 1.2, for all n such that $\frac{g_n}{g_\infty} \le 0.9$. But as n becomes larger *beyond the point where most of the opportunistic capacity has already been harvested*, the sub-optimality of OPS over SRPT starts to emerge and the ratio quickly deteriorates.

To summarize: *the presence of opportunistic capacity gains mitigates the sub-optimality of OPS, however, if load increases to a point where opportunistic capacity function saturates, OPS can become significantly sub-optimal.*

This observation naturally leads us to investigate a threshold-based or regulated-admission scheduler described in the next section – for all sufficiently large thresholds, the scheduler's competitive ratio is better than that of OPS.

E. SRPT-OPS scheduler for capacity regions $\boldsymbol{\mathcal{C}}$ and $\bar{\boldsymbol{\mathcal{C}}}$

The SRPT-OPS(n^*) scheduler admits to service at most a fixed number n^* of users according to SRPT, and serves the admitted users according to OPS. The details are as follows.

Definition 3: Consider a system with a concave increasing opportunistic capacity function $(g_k, k \ge 0)$ and the associated capacity regions $(\mathcal{C}_k, k \ge 0)$ or $\big(\bar{\mathcal{C}}_k, k \ge 0\big)$. Set a threshold $n^* > 0$. At any time t when there are $|q(t)| = n$ files in the system, SRPT-OPS(n^*) serves each of the *shortest* $\min(n, n^*)$ files at rate $\frac{g_{\min(n,n^*)}}{\min(n,n^*)}$, while the remaining $(n-n^*)^+$ files wait in a queue.

Lemma 3: For any concave increasing capacity function $(g_k,\ k \ge 0)$, and initial state $\big(q(0); (l_k(0),\ 1 \le k \le |q(0)|)\big)$, the cost under SRPT-OPS(n^*), $c_{sp}(\cdot)$, with threshold parameter $n^* \ge 1$ is given by,

$$c_{sp}\big(\boldsymbol{l}(0); |q(0)|\big) = \sum_{k=1}^{|q(0)|} \hat{\pi}_k l_{(k)}(0) \ , \tag{11}$$

where, for any $k > 0$,

$$\hat{\pi}_k \equiv \pi_{[k]} + \left\lfloor \frac{k-1}{n^*} \right\rfloor \frac{n^*}{g_{n^*}} \ , \tag{12}$$

where $[k] \equiv k \mod n^*$ (with $\pi_0 \equiv \pi_{n^*}$, see (8)) and $\left\lfloor \frac{k-1}{n^*} \right\rfloor$ is the integer part of $\frac{k-1}{n^*}$.

In particular, we note that for multiples for n^*, we have that, $\hat{\pi}_{kn^*} = \pi_{n^*} + \frac{(k-1)n^*}{g_{n^*}}$, which is affine in k with slope $\frac{n^*}{g_{n^*}}$. From (9), we have that for $g_k = \frac{1-a^k}{1-a}$, the coefficient θ_{kn^*} is affine in k with slope $\frac{n^*}{g_\infty}$. Therefore, the ratio $\frac{\hat{\pi}_{kn^*}}{\theta_{kn^*}}$ is monotone in k and $\sup_{k \ge 1} \frac{\hat{\pi}_{kn^*}}{\theta_{kn^*}} = \max\left(\frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}}\right)$. This leads to the following result.

Theorem 3: For some fixed $a \in [0,1)$, let $g_k = \frac{1-a^k}{1-a}$ for all $k \ge 0$. Let $c_s(\boldsymbol{l}; n)$ and $c_{sp}(\boldsymbol{l}; n)$ be the costs under SRPT-HPR and SRPT-OPS(n^*) respectively when starting with n files of sizes given in $\boldsymbol{l} \equiv (l_1, \ldots, l_n)$. Then we have that,

$$\sup_{n \ge 1, \boldsymbol{l} \in (0,\infty)^n} \frac{c_{sp}(\boldsymbol{l}; n)}{c_s(\boldsymbol{l}; n)} \ \le \ \max\left(\frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}}\right) \ . \tag{13}$$

Moreover, for any fixed $\epsilon > 0$,

$$\lim_{n \to \infty} \sup_{\boldsymbol{l} \in (\epsilon,\infty)^n} \frac{c_{sp}(\boldsymbol{l}; n)}{c_s(\boldsymbol{l}; n)} \ = \ \frac{g_\infty}{g_{n^*}} \ . \tag{14}$$

See Fig. 2 (right) and recall the discussion from Section V-D: we noted that for n such that $\frac{g_n}{g_\infty} \lesssim 0.9$ (or $\frac{g_\infty}{g_n} \gtrsim 1.1$), the competitive ratio of OPS (and implicitly the ratio $\frac{\pi_n}{\theta_n}$) was fairly small, *i.e.*, under 1.25. Therefore, in light of the bound "$\max\left(\frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}}\right)$" given in (13), if the admission threshold n^* is set at a point where most of the opportunistic capacity, say 90%, is useable under SRPT-OPS(n^*), then SRPT-OPS(n^*) will have fairly small competitive ratio. But unlike SRPT-HPR which allocates service from a fictitious larger capacity region $\bar{\mathcal{C}}_n$, the SRPT-OPS scheduler allocates service from the actual region $\mathcal{C}_n$ and is therefore feasible in a real system.

In fact, since $\frac{\pi_k}{\theta_k} \le 2$, for any n^* such that $\frac{g_\infty}{g_{n^*}} \le 2$, *i.e.*, at least half the opportunistic capacity is useable under SRPT-OPS(n^*), we have that the competitive ratio of SRPT-OPS(n^*) is always better than that of OPS. Moreover, by (14), as $n \to \infty$, the competitive ratio of SRPT-OPS becomes $\frac{g_\infty}{g_{n^*}}$.

Next, we discuss simulation results for a dynamic system.

VI. Dynamic system in steady state

In this section, we simulate the $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ OPS, SRPT-HPR, and SRPT-OPS(n^*) queues for lognormal file-size distribution with mean and variance of 1 and 1.7 respectively. Recall that $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ OPS and $M/GI/\boldsymbol{\mathcal{C}}$ OPS are indeed identical, and so are $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ SRPT-OPS(n^*) and $M/GI/\boldsymbol{\mathcal{C}}$ SRPT-OPS(n^*). Moreover, for the $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ OPS queue, various quantities of interest including the mean sojourn time can also be analytically computed, see, *e.g.*, [7, Proposition 3.1].

Fig. 3 shows the simulation results for polymatroid capacity regions $\bar{\boldsymbol{\mathcal{C}}}^{(1)}$ and $\bar{\boldsymbol{\mathcal{C}}}^{(2)}$ corresponding to two unbounded capacity functions $\boldsymbol{g}^{(1)} \equiv \left(g_k^{(1)}, k \ge 0\right)$ and $\boldsymbol{g}^{(2)} \equiv \left(g_k^{(2)}, k \ge 0\right)$, given by

$$g_k^{(1)} \ = \ \log_2(1+k) \quad \text{and} \quad g_k^{(2)} \ = \ \log_2(1+\log_2(1+k)).$$

Fig. 4 shows the simulation results for capacity regions $\bar{\boldsymbol{\mathcal{C}}}^{(3)}$ corresponding to a bounded capacity function $\boldsymbol{g}^{(3)} \equiv \left(g_k^{(3)}, k \ge 0\right)$, given by $g_k^{(3)} = 2\left(1 - \left(\frac{1}{2}\right)^k\right)$.

For $\boldsymbol{g}^{(1)}$ and $\boldsymbol{g}^{(2)}$, the ratio of *mean* sojourn time under OPS to that under SRPT-HPR stays around 1.05 and 1.2 respectively for $\boldsymbol{g}^{(1)}$ and $\boldsymbol{g}^{(2)}$. For $\boldsymbol{g}^{(3)}$, as the load increases, the ratio of *mean* sojourn time under OPS to that under SRPT-HPR increases sharply from 1.2 to 2. Whereas, the ratio of *mean* sojourn time under SRPT-OPS(5) to that under SRPT-HPR stays under 1.25. Also, see Fig. 5 for a plot of the mean sojourn time versus the threshold n^* for other values of threshold besides 5, indicating that the performance of SRPT-OPS is better than OPS for a wide range of thresholds.

After a caveat, we state our conclusions from these simulation results. As per Theorem 1, SRPT-HPR is optimal only for the *transient* system with polymatroid capacity regions $\bar{\boldsymbol{\mathcal{C}}}$. The optimal scheduler for the dynamic system $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ is not known, however, one can expect SRPT-HPR to be close to optimal. Therefore, formally speaking, the mean

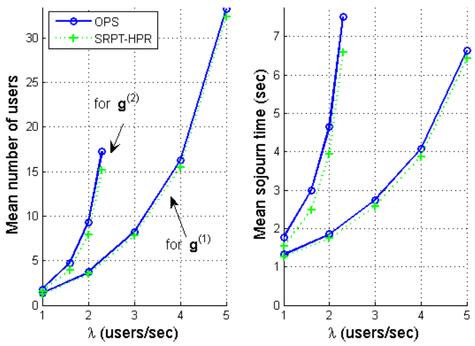

Fig. 3. Mean number of users in the system and mean sojourn time for unbounded capacity functions $\boldsymbol{g}^{(1)}$ and $\boldsymbol{g}^{(2)}$.

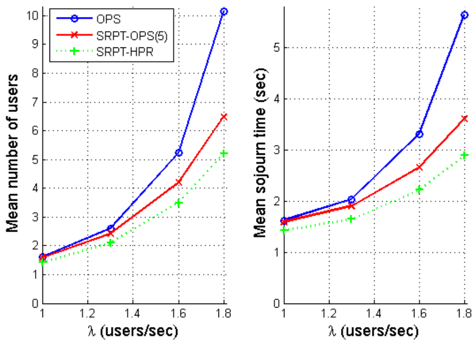

Fig. 4. Mean number of users in the system and mean sojourn time for bounded capacity function $\boldsymbol{g}^{(3)}$.

sojourn time under SRPT-HPR cannot be claimed as the minimum achievable mean sojourn time in $M/GI/\bar{\boldsymbol{\mathcal{C}}}$ system, or a lower bound for achievable mean sojourn time in the $M/GI/\boldsymbol{\mathcal{C}}$ system. Nevertheless, subsequently we will compare the performance of various schedulers against that of SRPT-HPR, as done for the transient system.

Now we are ready to state our conclusions from the results in Figs. 3–5. We conclude that for servers/channels with *high* marginal opportunistic gain (*e.g.*, $\log^2(n)$), a file-size aware scheduler will not offer a significant reduction in mean sojourn time over that of the file-size oblivious OPS (or SRPT-OPS with large threshold). Therefore, the only case of interest which offers room for improvement is where the channels exhibit only moderate marginal opportunistic gains (*e.g.*, $1 - a^n$). In this case, SRPT-OPS *with an appropriate threshold* can offer significant improvement over OPS, *e.g.*, for the shown results, SRPT-OPS is not more than 25% worse than SRPT-HPR. See [17] for further simulations and discussion.

VII. Conclusion

We presented new results and insights regarding the key tradeoff involved in scheduling best effort flows over time varying channels. Just as importantly, we reduced the complex scheduling problem to a simpler system, namely queue $M/GI/\boldsymbol{\mathcal{C}}$, which explicitly captures the tradeoff and other salient features of the original scheduling problem. For a given system load (number of users in the system), if the opportunistic capacity does not saturate, then the sub-optimality of file-size oblivious OPS (or SRPT-OPS with large threshold) is only minimal. If however the opportunistic capacity saturates at the given load, then OPS can be significantly sub-optimal,

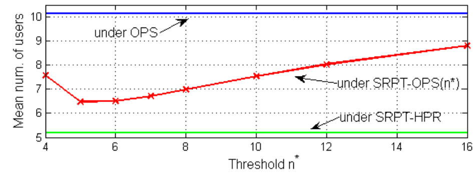

Fig. 5. Mean sojourn times under SRPT-OPS(n^*), OPS, and SRPT-HPR for capacity function $\boldsymbol{g}^{(3)}$ and user arrival rate $\lambda = 1.8$ user/sec.

whereas, SRPT-OPS, which makes use of file-size information, can still offer good performance. SRPT-OPS is simple to implement, reduces the amount of channel state feedback from users to the base station, and offers service at a nearly constant rate to the *admitted* users.

References

[1] L. Schrage, "A proof of the optimality of the Shortest Remaining Processing Time discipline," *Opns. Res.*, vol. 16, pp. 687–690, 1968.

[2] R. Knopp and P. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. IEEE Int. Conf. on Comm.*, vol. 1, Jun 1995, pp. 331–335.

[3] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Infomation Theory*, vol. 48, no. 6, pp. 1277–1294, Jun 2002.

[4] 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.

[5] D. Park, H. Seo, H. Kwon, and B. G. Lee, "A new wireless packet scheduling algorithm based on the CDF of user transmission rates," in *Proc. IEEE Glob. Telecom. Conf.*, vol. 1, Dec. 2003, pp. 528–532.

[6] S. Patil and G. de Veciana, "Managing resources and quality of service in heterogeneous wireless systems exploiting opportunism," *IEEE/ACM Trans. Networking*, vol. 15, no. 5, pp. 1046–1058, 2007.

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

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

[9] B. Tsybakov, "File transmission over wireless fast fading downlink," *IEEE Trans. Infomation Theory*, vol. 48, no. 8, pp. 2323–2337, 2002.

[10] M. Hu, J. Zhang, and J. Sadowsky, "Traffic aided opportunistic scheduling for wireless networks: algorithms and performance bounds," *Comput. Netw.*, vol. 46, no. 4, pp. 505–518, 2004.

[11] S. Aalto and P. Lassila, "Impact of size-based scheduling on flow-level performance in wireless downlink data channels," in *Proc. of ITC*, 2007.

[12] P. Lassila and S. Aalto, "Combining opportunistic and size-based scheduling in wireless systems," in *Proc. of MSWiM*, 2008.

[13] P. van de Ven, S. Borst, and S. Shneer, "Instability of maxweight scheduling algorithms," in *Proc. IEEE INFOCOM*, April 2009.

[14] B. Sadiq and G. de Veciana, "Throughput-optimality of *delay*-driven MaxWeight scheduler for a wireless system with flow dynamics," in *Proc. of Allerton Conf. on Comm. Control and Comp.*, Sep 2009.

[15] S. Liu, L. Ying, and R. Srikant, "Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics," in *Proc. IEEE INFOCOM*, 2010.

[16] E. M. Yeh and A. S. Cohen, "Delay optimal rate allocation in multiaccess fading communications," in *Proc. of Allerton Conf. on Comm. Control and Comp.*, 2004, pp. 140–149.

[17] B. Sadiq and G. de Veciana, "Balancing SRPT prioritization vs opportunistic gain in wireless systems with flow dynamics," *Technical Report*, 2010. [Online]. Available: www.ece.utexas.edu/~gustavo/SRPTvsOppG.pdf

[18] M. L. Pinedo, *Scheduling: theory, algorithms, and systems*, 3rd ed. Prentice-Hall, 2008.

[19] T. Gonzalez, "Minimizing the mean and maximum finishing time on uniform processors," *Technical Report CS–78–22, The Pennsylvania State University*, 1978.