The Impact of Queue Length Information on Buffer Overflow in Parallel Queues

Krishna Jagannathan, *Member, IEEE*, and Eytan Modiano, *Fellow, IEEE*

***Abstract*—We consider a system consisting of N parallel queues, served by one server. Time is slotted, and the server serves one of the queues in each time slot, according to some scheduling policy. We first characterize the exponent of the buffer overflow probability and the most likely overflow trajectories under the Longest Queue First (LQF) scheduling policy. Under statistically identical arrivals to each queue, we show that the buffer overflow exponents can be simply expressed in terms of the total system occupancy exponent of m parallel queues, for some $m \leq N$. We next turn our attention to the rate of queue length information needed to operate a scheduling policy, and its relationship to the buffer overflow exponents. It is known that queue length blind policies such as processor sharing and random scheduling perform worse than the queue aware LQF policy, when it comes to buffer overflow probability. However, we show that the overflow exponent of the LQF policy can be preserved with arbitrarily infrequent queue length updates.**

***Index Terms*—Buffer overflow probability, large deviations, queue length-based scheduling.**

I. Introduction

Scheduling is an essential component of any queueing system where the server resources need to be shared between many queues. Perhaps the most basic requirement of a scheduling algorithm is to ensure the stability of all queues in the system, whenever feasible. Much research work has been reported on "throughput optimal" scheduling algorithms that achieve stability over the entire capacity region of a network [11], [16]. While stability is an important and necessary first-order metric, most practical queueing systems have more stringent quality of service requirements.

In this paper, we consider a system consisting of N parallel queues and a single server. A scheduling policy decides which of the queues gets service in each time slot. Our aim is to better understand the relationship between the amount of queue length information required to operate a scheduling policy and the corresponding buffer overflow probability. The scheduling decisions may take into account the current queue lengths in the system, in which case we will call the policy "queue aware." If the scheduling decisions do not depend on the current queue lengths, except to the extent of knowing whether or not a queue is empty, we will call it a "queue-blind" scheduling policy.

We analyze the buffer overflow probability of the widely studied Longest Queue First (LQF) policy in the large-buffer regime. We assume that the queues are fed by statistically identical arrival processes. Under such a symmetric traffic pattern, we show that the large deviation exponent of the buffer overflow probability under LQF scheduling is expressible purely in terms of the total system occupancy exponent of an m queue system, where $m \leq N$ is determined by the input statistics. We also characterize the most likely overflow trajectories and show that there are at most N possible overflow modes that dominate.

Although any work-conserving policy [such as LQF, processor sharing (PS) or random scheduling (RS)] will achieve the same throughput region and total system occupancy distribution, the LQF policy outperforms queue-blind policies in terms of the buffer overflow probability. Equivalently, this implies that the buffer requirements are lower under LQF scheduling than under queue-blind scheduling, if we want to achieve a given overflow probability. For example, our study indicates that under Bernoulli and Poisson traffic, the buffer size required under LQF scheduling is only about 55% of that required under RS, when the traffic is relatively heavy. On the other hand, with LQF scheduling, the scheduler needs queue length information in *every* time slot, which leads to a significant amount of control signaling. Motivated by this, we identify a "hybrid" scheduling policy, which achieves the same buffer overflow exponent as the LQF policy, with arbitrarily infrequent queue length information.

A. Related Work

To our knowledge, Bertsimas *et al.* [1] were among the first to analyze the large deviations behavior of scheduling policies in parallel queues. They consider the case of two parallel queues and characterize the buffer overflow exponents under two important service disciplines, namely generalized processor sharing (GPS) and generalized LQF. We also refer to the related papers [12], [20], [22], where the authors analyze a system of parallel queues, with deterministic arrivals and time-varying connectivity. For a survey of asymptotics under PS, see [3].

Stolyar and Ramanan [14] study large deviations for the largest weighted delay first policy and prove the optimality of the exponent of the weighted delay under that policy. The single-node result was later generalized to a multiclass model with fixed routes in [13]. In a very similar spirit and setting,

Manuscript received February 09, 2010; revised August 11, 2011; accepted November 07, 2011. Date of publication June 14, 2013; date of current version September 11, 2013. This work was supported in part by NSF Grants CNS-0915988, CNS-0626781, and CNS-1217048, and in part by ARO MURI Grant W911NF-08-1-0238.

K. Jagannathan is with the Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600036, India (e-mail: krishnaj@ee.iitm.ac.in).

E. Modiano is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: modiano@mit.edu).

Communicated by R. D. Yates, Associate Editor for Communication Networks.

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

Digital Object Identifier 10.1109/TIT.2013.2268926

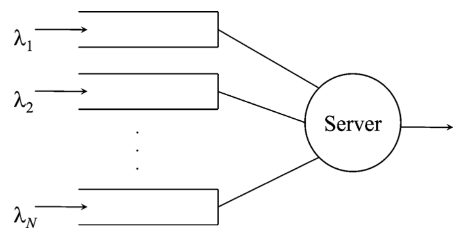


Fig. 1. N parallel queues served by one server.

Venkataramanan *et al.* [21] deal with large deviations of total end-to-end buffer overflow probability. Subramanian [15] derives large deviations of max-weight scheduling for general convex rate regions, rather than the simplex rate regions in [13] and [14]. In each case, the optimal exponent and the most likely overflow trajectory are obtainable by solving a variational control problem. In many cases, the optimal solution to the variational problem can be found by solving a finite-dimensional optimal control problem [1], [14], [15]. In [18] and [19], an interesting link is established between large deviation optimality and Lyapunov drift minimizing scheduling policies.

The rest of this paper is organized as follows. In Section II, we present the system description, and some preliminaries on large deviations. Our characterization of the large deviation behavior of LQF scheduling is presented in Section III. Section IV compares LQF scheduling to queue-blind scheduling in terms of the overflow probability and buffer scaling. In Section V, we study scheduling with infrequent queue length information. A shorter version of this paper appeared in [8].

II. System Description and Preliminaries

Fig. 1 depicts a system consisting of N parallel queues, served by one server. We assume that time is slotted, and the server is capable of serving one packet per slot. Within each queue, packets are served on a first come, first served basis. Arrivals occur according to a random process $A_i[t], i = 1, \ldots, N$, which denotes the number of packets that arrive at queue i during slot t. The arrivals to the different queues are independent. We assume a symmetric traffic pattern, i.e., the arrival processes to each queue are statistically identical to each other. For simplicity, let us also assume that the arrivals are independent across time slots. The average arrival rate to a queue is $\mathbb{E}[A_i[t]] = \lambda$ packets/slot for each i. For stability, we assume that the condition $\lambda < \frac{1}{N}$ is satisfied. Let us also define

$$A_i[t_1, t_2] = \sum_{\tau=t_1}^{t_2} A_i[\tau], \quad t_1 \leq t_2$$

as the number of arrivals to queue i between time slots t_1 to t_2. The queue evolution equation is given by

$$Q_i[t+1] = \max(0, Q_i[t] + A_i[t] - S_i[t]), \quad i = 1, 2. \ldots, n, \tag{1}$$

where $S_i[t]$ denotes the service allocated to queue i during slot t.

The log-moment generating function of the input process to each queue, defined by

$$\Lambda(\theta) = \log \mathbb{E}[\exp(\theta A_i[1])]$$

is assumed to exist for *each* $\theta > 0$. The *convex dual* of $\Lambda(\theta)$ is defined by

$$\Lambda^*(x) = \sup_{\theta}[\theta x - \Lambda(\theta)].$$

It is well known from the strong law of large numbers that the sequence of empirical means $A_i[1,n]/n, \ n \in \mathbb{N}$ converges almost surely to λ. Cramér's theorem [6, Th. 1.3] shows that the sequence of empirical means satisfies an LDP with rate function $\Lambda^*(\cdot)$ on the real line. Specifically, for any set $G \subset \mathbb{R}$ with interior G^o and closure $\overline{G}$, we have

$$-\inf_{x \in G^o} \Lambda^*(x) \leq \liminf_{n \to \infty} \frac{1}{n} \log \mathbb{P}\left\{\frac{A_i[1,n]}{n} \in G\right\} \leq \limsup_{n \to \infty} \frac{1}{n} \log \mathbb{P}\left\{\frac{A_i[1,n]}{n} \in G\right\} \leq -\inf_{x \in \overline{G}} \Lambda^*(x).$$

Intuitively of course, this implies that the probability of the sample mean deviating from λ decays exponentially in n, where the rate of decay is given by $\Lambda^*(\cdot)$.

Next, let us consider the sample paths of the sequence of partial sums process

$$A_i^n(t) = \frac{1}{n} A_i[1, \lfloor nt \rfloor], \quad t \in [0,1]. \tag{2}$$

In effect, we have scaled both time and vertical axes by n, and embedded the arrival sequence into *càdlàg* functions (i.e., right-continuous functions with left limits) on the interval $[0,1]$. Again, for the sequence of processes $A_i^n(t), \ n \in \mathbb{N}$, a *functional* strong law of large numbers can be shown to hold. That is, the sequence $A_i^n(t), \ n \in \mathbb{N}$ converges almost surely to the deterministic process $\lambda t, \ t \in [0,1]$, on the space of *càdlàg* functions endowed with the supremum norm; see [7, Th. 4].

Analogously to Cramér's theorem, we may expect a large deviation bound on the probability that the sequence of sample paths deviates from the almost sure limit $\lambda t, \ t \in [0,1]$. Such a result does exist, although a precise formulation of such an LDP requires defining a suitable topology on the space of *càdlàg* functions on $[0,1]$. A sample path LDP for the partial sums process, first shown by Varadhan [17], uses the supremum norm topology, while subsequent generalizations work with the Skorokhod topology [2], [9], [10]. In particular, the sequence $A_i^n(\cdot), \ n \in \mathbb{N}$, satisfies an LDP on the space of *càdlàg* functions on $[0,1]$ equipped with the supremum norm, with rate function given by

$$I(a(t)) = \int_0^1 \Lambda^*(\dot{a}(t))dt, \tag{3}$$

if $a(t)$ is absolutely continuous and $a(0) = 0$. If not, the rate function is infinite [5, Th. 5.1.2].

A sample path LDP of the form given in (3) may be satisfied under more general conditions than independent and identically distributed (i.i.d.) arrivals in each time slot. In [4, Th. 5], general mixing conditions for a stationary increment sequence to satisfy a sample path LDP of the form in (3) are derived. Although we have assumed i.i.d. arrivals for simplicity, all we really need is the arrival process to satisfy an LDP with rate function $I(\cdot)$, along with the scaling condition

$$\lim_{x \to \infty} \frac{1}{x} \Lambda^*(x) = \infty.$$

In the i.i.d. case, the above scaling condition is implied by our assumption that $\Lambda(\theta) < \infty$ for each $\theta > 0$.

We are interested in the steady-state probability of a buffer overflow in the large-buffer regime, under a given scheduling policy Π. Let us denote the overflow limit by M. More specifically, we are interested in the exponent of the overflow probability under the large-buffer scaling, which is defined[1] as

$$E_N^{\Pi} = \lim_{M\to\infty} -\frac{1}{M}\log \mathbb{P}\left\{\max_{i=1,\dots,N} Q_i[0] \geq M\right\}, \tag{4}$$

where $Q_i[0]$ denotes the steady-state queue length, assuming the system has been running for a long time. We emphasize that this exponent depends on the scheduling policy Π, as well as the system size N and the input statistics. We also define the exponent corresponding to the total system occupancy exceeding a certain limit:

$$\Theta_N = \lim_{q\to\infty} -\frac{1}{q}\log \mathbb{P}\left\{\sum_{i=1}^{N} Q_i[0] \geq q\right\}. \tag{5}$$

As we shall see, the system occupancy exponent in (5) plays an important role in our analysis of the buffer overflow exponent (4) under the LQF policy. The limit in (5) is well defined for all work-conserving policies. Indeed, the following well-known lemma asserts that Θ_N is the same for all *work-conserving* scheduling policies.

Lemma 1: All work-conserving policies achieve the same steady-state system occupancy distribution and, hence, the same system exponent Θ_N.

In fact, the above result holds at a sample-path level, since one packet would leave the system every time slot if the system is not empty, under any work-conserving policy.

We first analyze the LQF scheduling policy, which, as the name implies, serves the longest queue in each slot, with an arbitrary tie-breaking rule. We also consider two other work-conserving policies: RS, which serves a random occupied queue in each slot (each with equal probability), and PS, which divides the server capacity equally between all occupied queues. Note that LQF scheduling is queue-aware, while RS and PS are queue-blind.

III. Buffer Overflow Probability Under LQF Scheduling

In this section, we present our main results regarding the buffer overflow exponents and trajectories under LQF scheduling. We begin by characterizing the system occupancy exponent Θ_N for a work-conserving policy.

Proposition 1: Under any work-conserving policy, the system occupancy exponent is given by

$$\Theta_N = \inf_{a>0} \frac{1}{a}\Lambda^*(a + \frac{1}{N}). \tag{6}$$

Proof (Outline): The result is a consequence of the fact that the total system occupancy distribution is the same as the queue length distribution of a single queue, served by the same server, but fed by the sum process $\sum_i A_i(t)$. Since the input processes to the different queues are i.i.d., the log-moment generating function of the sum process is $N\Lambda(\theta)$. Next, from the definition of the convex dual, the rate function of the sum process can be expressed as $N\Lambda^*(x/N)$. Once the rate function of the input process is known, the overflow exponent of a single-server queue can be easily computed. In particular, the result is a simple consequence of [6, Th. 1.4]. □

We now define scaled processes for the arrivals and queue lengths, which are often used to study sample path large deviations in the large-buffer regime. For every sample path that leads to a buffer overflow at time slot 0, there exists a time $-n \leq 0$ for which all queues are empty. Since we are interested in large M asymptotics, we let $T = \frac{n}{M}$, and define the sequence of scaled queue length processes

$$q_i^{(M)}(t) = \frac{Q_i[\lfloor Mt \rfloor]}{M}, \quad i = 1, \dots, N. \tag{7}$$

These are *càdlàg* functions defined on the interval $[-T, 0]$. Similarly, we define a scaled version of the cumulative arrival processes[2]

$$A_i^{(M)}(t) = \frac{A_i[-MT, \lfloor Mt \rfloor]}{M}, \quad t \in [-T, 0].$$

The initial condition implies that $q_i(-T) = 0, i \leq N$. Under the above scaling, $q_i(0) \geq 1$ corresponds to the overflow of queue i at time 0.

Analogously to (3), we assume that the sequence $A_i^{(M)}(t)$ satisfies an LDP in the space of *càdlàg* functions on $[-T, 0]$ endowed with the uniform topology, with rate function given by

$$I_T(a_i(t)) = \int_{-T}^{0} \Lambda^*(x_i(t))dt, \tag{8}$$

when $a(t)$ is absolutely continuous and $a(-T) = 0$. The rate function is infinite otherwise. In (8), $x_i(t)$ is the almost everywhere derivative of the absolutely continuous function $a_i(t)$ and, hence, has the interpretation of the empirical rate of the arrival process to queue i. In particular, the assumption that $A_i^{(M)}(t)$ satisfies an LDP is automatically satisfied for i.i.d. arrivals as discussed earlier.

Next, in order to specify the queue evolution under LQF scheduling, let us partition $\mathbb{R}_+^N$ as follows. Let $\mathcal{I}$ be any nonempty subset of $\{1, 2, \dots, N\}$. We define $\mathcal{R}_{\mathcal{I}}$ such that $(q_1, \dots, q_N) \in \mathcal{R}_{\mathcal{I}}$ iff $q_i = q_j \ \forall i, j \in \mathcal{I}$, and for any $k \notin \mathcal{I}, \ j \in \mathcal{I}, \ q_k < q_j$. Intuitively, in region $\mathcal{R}_{\mathcal{I}}$, the queues in the index set $\mathcal{I}$ evolve together, and all other queues are smaller. For example, in the case $N = 2$, the positive quadrant is partitioned into three regions, namely $\mathcal{R}_1 = \{(q_1, q_2) \,|\, q_1 > q_2\}$, $\mathcal{R}_2 = \{(q_1, q_2) \,|\, q_2 > q_1\}$, and $\mathcal{R}_{1,2} = \{(q_1, q_2) \,|\, q_1 = q_2\}$. It is clear that the regions $\mathcal{R}_{\mathcal{I}}$ are convex and constitute a partition of $\mathbb{R}_+^N$ as $\mathcal{I}$ ranges over all nonempty index sets.

[1]The limit in (4) may not exist for an arbitrary policy Π. Thus, to be precise, one should consider the corresponding $\liminf$ and $\limsup$, which always exist as long as the overflow event is measurable under policy Π.

[2]This is an abuse of notation when compared to (2). We define these processes in the interval $[-T, 0]$ instead of $[0, 1]$ in order to study an overflow event at time 0.

When the vector of scaled queue lengths (7) lies in the region $\mathcal{R}_{\mathcal{I}}$, the evolution relation is given by

$$\sum_{i\in\mathcal{I}} \dot{q}_i(t) = \sum_{i\in\mathcal{I}} x_i(t) - 1;$$
$$\dot{q}_k(t) = x_k(t),\ \forall k \notin \mathcal{I}, \tag{9}$$

where the dotted quantities denote derivatives.[3]

We now state the main result regarding the large deviations behavior of LQF scheduling. The result states that the exponent under LQF scheduling is expressible purely in terms of the system occupancy exponents. Recall from Proposition 1 that the system occupancy exponent of j parallel queues is given by

$$\Theta_j = \inf_{a>0} \frac{1}{a}\Lambda^*(a + \frac{1}{j}). \tag{10}$$

Define a_j^* to be the optimizing value of a in (10). Let us define the candidate index set $\mathcal{I}^* \subseteq \{1, 2, \ldots, N\}$, as

$$\mathcal{I}^* = \left\{ j \,\middle|\, 1 \le j < N,\ a_j^* > \lambda \right\} \cup \{N\}. \tag{11}$$

We remark that $\mathcal{I}^*$ consists of only those indices $j \le N-1$ for which the likeliest overflow rate a_j^* exceeds the arrival rate λ; however, $\mathcal{I}^*$ always includes the index N regardless of the value of a_N^*. The interpretation of this will be clear once the likeliest overflow trajectories predicted by Theorem 1 are understood.

Theorem 1: Under independent and statistically identical arrival processes to each queue, the large deviation behavior of buffer overflow under LQF scheduling is given as follows:

i) The exponent is given by

$$E_N^{\mathrm{LQF}} = \min_{j\in\mathcal{I}^*} j\Theta_j, \tag{12}$$

where Θ_j is the system occupancy exponent for j parallel queues, given by (10).

ii) For a given λ, suppose that a unique $k \le N$ minimizes (12). If $k < N$, the most likely overflow event consists of k queues reaching overflow, and the remaining queues growing to $O(M)$ without overflowing. If $k = N$, the most likely overflow event consists of all queues reaching overflow together.

The proof of the theorem involves invoking an LDP for the queue length process, and solving a variational problem to obtain the rate function. We relegate the proof to Appendix A and discuss the result intuitively.

The first part of the theorem states that the buffer overflow exponent under LQF scheduling is only a function of the system occupancy exponent Θ_j of a system with j parallel queues, where $j \in \mathcal{I}^*$. The candidate index set $\mathcal{I}^*$ consists of all j for which $a_j^* > \lambda$ for $j \le N-1$, and always includes N. The second part of the theorem asserts that if E_N^{LQF} equals $k\Theta_k$ for a unique $k < N$, then the most likely overflow scenario consists of k queues overflowing, and the other $N-k$ queues grow approximately to $M\frac{\lambda}{a_k^*}$, which is less than M. In particular, the queues that do not overflow are never the longest and, hence, get no service. The service is shared equally among the k queues that overflow, and a_k^* denotes the most likely *rate* at which the k queues overflow, in spite of getting all the service. On the other hand, the queues that do not overflow get to keep all their arrivals, which occur at the average rate λ. The exponent for this case is given by $k\Theta_k$, which corresponds to all the queues in a k-queue system overflowing together. This is because the $N-k$ queues which do not get service receive arrivals at the average rate, and hence do not contribute to the exponent. Finally, if $j = N$ minimizes (12), then all queues reach overflow together, while growing at rate a_N^*. In this case, note that there is no restriction on a_N^*, which explains why N is always included in the candidate index set $\mathcal{I}^*$.

[3]We use the notation of derivative for simplicity of notation, although the derivatives may not exist everywhere. Strictly speaking, we ought to write down these equations with integrals, as shown in [1].

A. Illustrative Examples With Bernoulli Traffic

In this section, we obtain the LQF exponents explicitly for a system with symmetric Bernoulli inputs to each queue. We deal with $N = 2$ and $N = 3$, since these cases are easily visualized and elucidate the nature of the solution particularly well. We begin by making the following elementary observation regarding LQF scheduling and Bernoulli arrivals

Proposition 2: Under Bernoulli arrivals and LQF scheduling, the system evolves such that the two longest queues never differ by more than two packets.

Next, we state a well-known result regarding the rate function $\Lambda^*(\cdot)$ for a Bernoulli process.

Proposition 3: For a Bernoulli process of rate λ, the rate function is given by

$$\Lambda^*(x) = D(x\|\lambda) := x\log\frac{x}{\lambda} + (1-x)\log\frac{1-x}{1-\lambda},$$

where $D(x\|\lambda)$ is the Kullback–Liebler divergence (or the relative entropy) between x and λ.

The result is a consequence of Sanov's theorem for finite alphabet [6, Th. 2.9].

Let us now consider a two-queue system with Bernoulli arrivals. For this simple system, it turns out that the exponent can be computed from first principles, without resorting to sample path large deviations. First, the system exponent Θ_2 under Bernoulli arrivals can be computed either from (6), or directly from the system occupancy Markov chain, yielding

$$\Theta_2 = 2\log\frac{1-\lambda}{\lambda}.$$

The overflow behavior under LQF scheduling is derived from first principles in the following proposition.

Proposition 4: Under LQF scheduling and Bernoulli arrivals, the following statements hold for the case $N = 2$:

i) The most likely overflow trajectory is along the *diagonal,* $(q_1 = q_2)$

ii) $E_2^{\mathrm{LQF}} = 2\Theta_2 = 4\log\frac{1-\lambda}{\lambda}$.

Proof: Part (i) of the result is a simple consequence of Proposition 2. Specifically, suppose that one of the queues (say Q_1) overflows, so that $Q_1 \ge M$. From Proposition 2, it follows that $Q_2 \ge M-2$. Thus, when an overflow occurs in one queue, the other queue is also about to overflow, so that the only possible (and thus the most likely) overflow trajectory is along the diagonal.

In order to show part (ii), we first argue that $E_2^{\text{LQF}} \geq 2\Theta_2$. Indeed, when a buffer overflow occurs, the total system occupancy is at least $2M-2$. Thus, the buffer overflow probability is upper-bounded by the probability of the total system occupancy being at least $2M-2$:

$$\mathbb{P}\{Q_1 \geq M\} \leq \mathbb{P}\{Q_1+Q_2 \geq 2M-2\}.$$

We thus have,

$$\begin{aligned} E_2^{\text{LQF}} &= \lim_{M\to\infty} -\frac{1}{M}\log \mathbb{P}\{Q_1 \geq M\} \\ &\geq \lim_{M\to\infty} -\frac{1}{M}\log \mathbb{P}\{Q_1+Q_2 \geq 2M-2\} = 2\Theta_2, \end{aligned}$$

where the last equality follows from the definition of Θ_2.

To show a matching upper bound, note that when the system occupancy is $2M$ or greater, at least one of the queues will necessarily overflow. Thus,

$$\mathbb{P}\{Q_1+Q_2 \geq 2M\} \leq \mathbb{P}\{\max(Q_1,Q_2) \geq M\}.$$

We can then argue as above that $E_2^{\text{LQF}} \leq 2\Theta_2$. □

Let us now analyze a system with three queues, fed by symmetric Bernoulli traffic. In this case, although the longest two queues grow together, it is not immediately clear how the third queue behaves during overflow. As before, the system occupancy exponent Θ_3 can be obtained from (6) or directly from the Markov chain to yield

$$\Theta_3 = \log \frac{2A}{(B-2C)+\sqrt{(B-2C)^2+4AC}}$$

where $A=(1-\lambda)^3$, $B=3\lambda^2$, and $C=\lambda^3$.

Let us use Theorem 1 to calculate the desired overflow exponent. Note first that Θ_1 is infinite in this case, since a single queue fed by Bernoulli inputs cannot overflow. Next, to decide whether or not 2 is a candidate index for a given λ, we see by direct computation that $a_2^* = 1/2-\lambda$. Thus, 2 is a candidate index only for $\lambda < 1/4$. Therefore, the exponent is $\min(2\Theta_2, 3\Theta_3)$ for $\lambda < 1/4$, and $3\Theta_3$ for $\lambda > 1/4$.

Fig. 2 shows a plot of $2\Theta_2$ and $3\Theta_3$ as functions of the input rate λ on each queue. It is clear from the figure that for small values of λ, the exponent $2\Theta_2$ dominates the overflow behavior. In this regime, the most likely manner of overflow involves two queues reaching overflow, while the third queue grows to approximately $M\frac{\lambda}{1/2-\lambda}$. For larger values of $\lambda(> 0.07)$, the exponent is $3\Theta_3$, and all three queues overflow together.

IV. LQF Versus Queue-Blind Policies

In this section, we compare the performance of LQF scheduling with that of queue-blind policies. We only consider a two-queue system, since the large deviation behavior of PS and RS is difficult to characterize for $N > 2$. The following result for PS follows from [1].

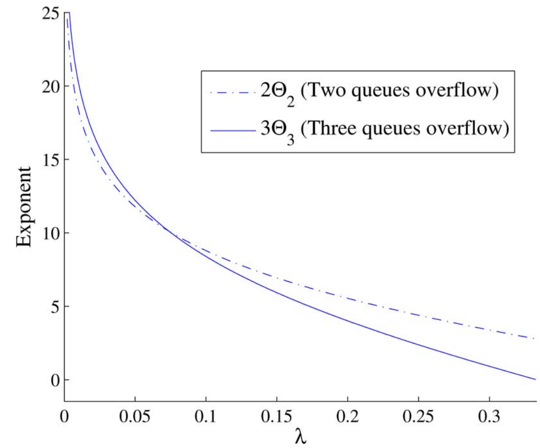


Fig. 2. Exponent behavior for $N=3$ under Bernoulli traffic.

Proposition 5: The buffer overflow exponent for a two-queue system under PS is given by

$$E_2^{\text{PS}} = \inf_{a>0} \frac{1}{a}\left[\Lambda^*(a+\frac{1}{2})+\Lambda^*(\frac{1}{2})\right]. \tag{13}$$

The most likely manner of overflow under PS is as follows. Suppose it is the first queue that overflows. The second queue receives traffic at rate 1/2, which is also its service rate. Thus, the second queue does not overflow, and grows to at most $o(M)$. The first queue receives service at rate 1/2 and input traffic at rate $a_{ps}^*+1/2$, where a_{ps}^* optimizes (13). Thus, a_{ps}^* is the most likely rate of overflow of the first queue.

Next, we present the exponent for RS.

Proposition 6: The buffer overflow exponent for a two-queue system under RS is given by

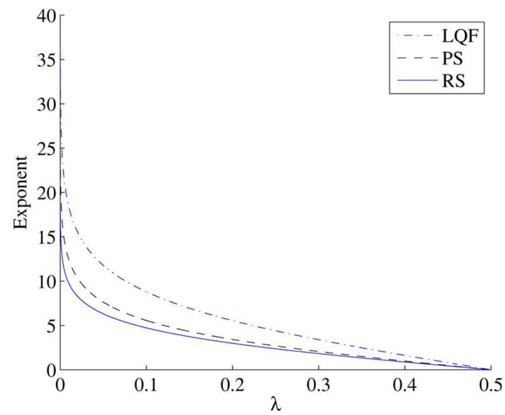
$$E_2^{\text{RS}} = \inf_{a>0}\frac{1}{a}\inf_{\phi\in(0,1)}\left[\Lambda^*(a+1-\phi)+\Lambda^*(\phi)+D(\phi\|\frac{1}{2})\right]. \tag{14}$$

The proof is outlined in Appendix B. We now describe the most likely overflow event. Suppose queue 1 overflows. The parameter ϕ that appears in the inner infimization in (14) denotes the empirical fraction of service received by queue 2. In other words, the "fair" coin tosses that decide which queue to serve when both queues are nonempty, "misbehave" statistically. The exponent corresponding to this event is given by $D(\phi\|\frac{1}{2})$. If ϕ^* is the optimal value of ϕ in (14), the second queue receives traffic at rate ϕ^* and, therefore, grows to an $o(M)$ level. The first queue receives traffic at rate $a_{rs}^*+1-\phi^*$, where a_{rs}^* is the optimizing value of a in (14).

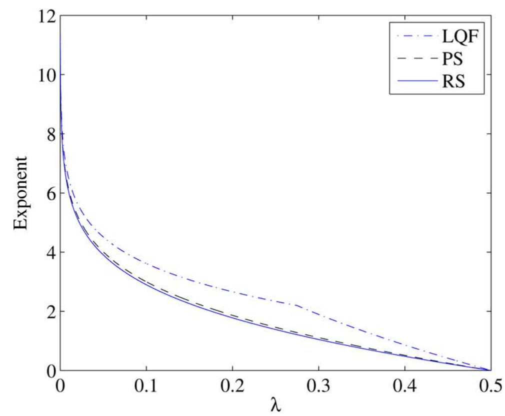
It has been shown that LQF has the best buffer overflow exponent among a fairly general class of scheduling policies [14]. We, therefore, expect the exponent under LQF scheduling to be larger than under queue-blind policies. The following result establishes the order among the overflow exponents for the three policies considered in this paper.

Proposition 7: It holds that $E_2^{\text{RS}} \leq E_2^{\text{PS}} \leq E_2^{\text{LQF}}$.

Proof: To see the first inequality $E_2^{\text{RS}} \leq E_2^{\text{PS}}$, note that substituting $\phi = 1/2$ into the RS exponent (14) yields the PS exponent. To prove the second inequality, it suffices to show that

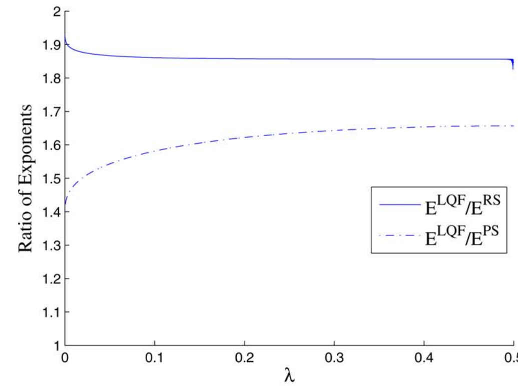


(a)

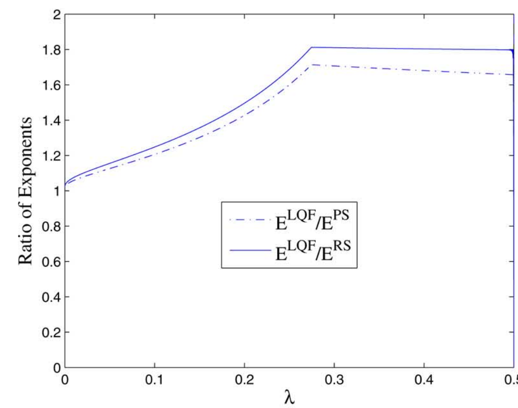


(b)

Fig. 3. Comparison of LQF, PS and RS exponents for a two-queue system, under (a) Bernoulli arrivals and (b) Poisson arrivals.



(a)



(b)

Fig. 4. Ratio of LQF exponent to PS and RS exponents for (a) Bernoulli arrivals and (b) Poisson arrivals.

$E_2^{\mathrm{PS}} \leq \Theta_1$ and $E_2^{\mathrm{PS}} \leq 2\Theta_2$. First note that for all $a \geq 0$, we have $\Lambda^*(a+1/2) \geq \Lambda^*(1/2)$ since the input rate λ is less than 1/2. Thus, for all $a \geq 0$,

$$\frac{2}{a}\Lambda^*(a+1/2) \geq \frac{1}{a}[\Lambda^*(a+1/2)+\Lambda^*(1/2)].$$

Taking infimum on both sides, we have $E_2^{\mathrm{PS}} \leq 2\Theta_2$. Similarly, for all $a > 0$, it can be shown that $\Lambda^*(a+1) \geq \Lambda^*(a+1/2) + \Lambda^*(1/2)$, using the fact that $\Lambda^*(\cdot)$ is an increasing convex function, for arguments greater than λ. Dividing the preceding inequality by a and taking infimum, it follows that $E_2^{\mathrm{PS}} \leq \Theta_1$. □

In Fig. 3, we plot the exponents corresponding to LQF, PS, and RS for a two-queue system, as a function of the arrivals rate λ. Fig. 3(a) corresponds to having Bernoulli arrivals in each time slot, while in Fig. 3(b), the number of arrivals in each slot is a Poisson random variable. The first observation we make from Fig. 3 is that, for a given arrival rate, the exponent values for a given policy are generally larger under Bernoulli traffic. This is because Poisson arrivals have a larger potential for being more bursty, and hence, the overflow probability is larger (and the exponent smaller) for a given average rate. Next, notice that the LQF exponent under Poisson traffic [see Fig. 3(b)] exhibits a cusp at $\lambda \approx 0.27$. This is because under Poisson traffic, we have two competing exponents Θ_1 and $2\Theta_2$, corresponding respectively to one queue and both queues overflowing. For λ below the cusp, Θ_1 dominates, and vice-versa. On the other hand, under Bernoulli traffic, Θ_1 is infinite. Thus, the LQF exponent is given by $2\Theta_2$, which is a smooth curve as shown in Fig. 3(a).

A. Buffer Scaling Comparison

It is well known that large deviation exponents have direct implications on the buffer size required in order to achieve a certain low probability of overflow. We now compare LQF scheduling with the two queue-blind policies in terms of the buffer scaling required to guarantee a given overflow probability.

In Fig. 4, we plot the ratio of the LQF exponent to the PS and RS exponents. This ratio is related to the savings in the buffer size that one can expect from using LQF scheduling, as opposed to using one of the queue-blind policies. For example, consider the ratio of the LQF exponent to the RS exponent, when the traffic is relatively heavy (say $\lambda > 0.3$). This is the regime where overflows are most likely to occur. We see that under both Bernoulli and Poisson traffic, the LQF exponent is roughly 1.8 times the RS exponent. In the very large-buffer regime, this indicates that in order to achieve a certain order of magnitude of overflow probability, the LQF policy requires only about 55% of the buffer size required under RS in heavy traffic. In realistic scenarios, however, the actual buffer savings may also be affected by pre-exponential terms, which are not captured by the LD exponents, and must be estimated numerically.

V. Scheduling With Infrequent Queue Length Information

We have seen that the LQF policy has a superior buffer overflow performance compared to queue-blind policies. This is because the queue-blind policies cannot discern and mitigate large build-up on one of the queues, whereas the LQF policy tries to achieve a more balanced set of queues by serving the longest queue in each slot. On the other hand, the scheduler needs to know queue length information in every slot in order to perform LQF scheduling. In this section, we will show that the buffer overflow performance of LQF scheduling can be maintained even if we allow for arbitrarily infrequent queue length information to be conveyed to the scheduler.

The basic idea is that it is sufficient to serve the longest queue only when the queues are large. When the queue lengths are all small, we can save on the queue length information by adopting a work-conserving, but queue-blind scheduling strategy. To achieve this, we suggest the following scheduling policy which is a "hybrid" version of the queue-blind RS, and the LQF policy.

Hybrid Scheduling: Let $K < M$ be a given queue length threshold. In each slot, if all queues are smaller than K, then serve any occupied queue at random. If at least one queue exceeds K, serve the longest queue in that slot.

The following theorem asserts that the hybrid policy achieves the same buffer overflow exponent as LQF scheduling, while requiring queue length information in a vanishingly small fraction of slots.

Theorem 2: For the hybrid scheduling policy proposed above, the following statements hold.

i) The fraction of slots in which queue length information is required can be made arbitrarily small.
ii) The buffer overflow exponent of hybrid scheduling is equal to E_N^{LQF}, as long as $K = o(M)$.

Observe that queue length information is required only in time slots when the longest queue in the system is longer than K. Since RS is a stabilizing policy, the steady-state probability that the longest queue exceeds K approaches zero as K becomes large. (In fact, this probability goes to zero *exponentially* in K.) Therefore, the fraction of slots in which queue length information is required can be made arbitrarily small. On the other hand, the overflow exponent remains the same as in the LQF case. This is because the hybrid policy differs from LQF scheduling only in a "small" neighborhood around the origin. We relegate the proof to Appendix C.

Remarks:

1) Throughout the paper, we have assumed that queue-blind policies know whether each queue is empty or not empty. This assumption ensures that the queue-blind policies are work-conserving. Indeed, it is easy to see that a scheduling policy which does not have any knowledge about empty queues will waste some time slots by allocating service to them.
2) In order to better understand the amount of queue length information required to operate a scheduling policy, consider a wireless uplink system with multiple users being served by a base station. To perform LQF scheduling, each user would have to quantize and encode its own queue length[4] and transmit it to the base station, along with the payload. This would require a variable number of bits per time slot depending on the longest queue. Further, since the queue lengths are unbounded, there is a need for a dynamic quantization scheme that necessitates further coordination between the nodes and the base station. Thus, LQF scheduling leads to significant control overheads. On the other hand, a single-bit suffices to obtain empty/nonempty information. In practice, this can be accomplished using an explicit reservation request packet, or a pilot symbol.
3) Along similar lines, in order to operate the hybrid policy, the base station can learn that the longest queue in the system is larger than the threshold K by having each user transmit a predesignated pilot symbol. After transmitting the pilot symbol, the subset of users with queue length greater than K can quantize and encode their queue length and transmit it to the base station. We have shown that the fraction of time slots when this queue length signaling must take place goes to zero under the hybrid policy. The problem of designing optimal quantization schemes for the users to report potentially unbounded queue lengths is an interesting problem for future work.

[4]Or equivalently, the queue length differential from the previous slot.

VI. Conclusions and Future Work

In this paper, we studied the buffer overflow probabilities in a system of parallel queues, under some well-known scheduling policies. We showed that under max-weight (or LQF) scheduling and symmetric traffic on all the queues, the large deviation exponent of buffer overflow probability is purely a function of the total system occupancy exponent. We also showed that queue length blind policies such as PS have a smaller overflow exponent (and hence larger buffer size requirements) than max-weight scheduling. Finally, we showed that the superior buffer overflow performance of LQF policy can be preserved even under arbitrarily infrequent queue length information. However, the problem of designing optimal quantization and encoding schemes for reporting potentially unbounded queue lengths is an interesting avenue for future work.

Appendix

1) Proof of Theorem 1: The proof consists of two parts. The first part involves showing that the queue length process under LQF scheduling satisfies an LDP, whose rate function is given by the solution to a variational problem. The second step involves solving the variational problem in the case of symmetric arrivals and proving that the optimal solution to the variational problem takes a simple form, as given by the theorem.

The existence of an LDP for the queue length process under max-weight like policies has been shown under fairly general conditions. For example, Stolyar and Ramanan [14] derives an LDP under longest weighted waiting time as well as longest weighted queue length scheduling. In [15], the LDP is extended to more general convex rate regions. The LDP for the queue length process can be shown on the space of *càdlàg* functions endowed with the uniform topology [14], or on a Skorokhod space as done in [15]. Once the LDP for the queue length vector is available, we can obtain the buffer overflow exponent by invoking a suitable *contraction principle*; see, for example, [15, Corollary 3.1].

Assuming without loss of generality that the first queue overflows, the exponent is given by the following variational problem:

$$\inf_{\mathbf{a}(t)\in\mathcal{AC},\mathbf{a}(-T)=0}\int_{-T}^{0}\left[\sum_{i=1}^{N}\Lambda^*(x_i(t))\right]dt \tag{15}$$

subject to

$$\begin{aligned} q_i(-T) &= 0, \forall i\\ q_1(0) &= 1,\\ T &: \text{free},\\ q_j(0) &: \text{free for } j>1, \end{aligned}$$

and the queue length trajectories $q_i(t)$ evolve according to (9). In (15), the minimization is over the set $\mathcal{AC}$ of all absolutely continuous trajectories such that $\mathbf{a}(-T)=0$, and $x_i(t)$ is the almost everywhere derivative of $a_i(t)$.

We focus on solving the above variational problem under the symmetric traffic scenario. In (15), the empirical rates $x_i(t)$ are the control variables, and the cost function is the exponent corresponding to the control variables, as given by Mogulskii's theorem (8). In words, the variational problem is to find the set of empirical rates which leads to the smallest exponent, and results in the overflow of at least one queue. Note that the above is a free time problem, i.e., the time T over which overflow occurs is not constrained. Also, it is possible for queues other than the first queue to reach overflow.

An important property which helps us solve (15) is given by the following lemma, which states that when the queue lengths are within one of the regions $\mathcal{R}_{\mathcal{I}}$, the empirical rates $x_i(t)$ can be taken as constants, without loss of optimality.

Lemma 2: Fix a time interval $[-T_1,-T_2]$ and consider a control trajectory $x_i(t),\ i=1,\dots,N,\ t\in[-T_1,-T_2]$, such that the fluid limit of the queue lengths $q_i(t),\ i=1,\dots,N,\ t\in[-T_1,-T_2]$ stay within a particular region $\mathcal{R}_{\mathcal{I}}$. Define the average control trajectory $\bar{x}_i$ in the interval $[-T_1,-T_2]$ as

$$\bar{x}_i(\tau)=\frac{1}{T_1-T_2}\int_{-T_1}^{-T_2}x_i(t)dt$$

for $i=1,\dots,N$ and $\tau\in[-T_1,-T_2]$. Then, the queue lengths under the average control trajectory $\bar{x}_i(t)$ lie entirely within $\mathcal{R}_{\mathcal{I}}$, and satisfy the same initial and final conditions at $t=-T_1$ and $t=-T_2$, respectively. Furthermore, the cost achieved under the (constant) control trajectory $\bar{x}_i(t)$ is not larger than the cost achieved under $x_i(t)$.

Proof: The proof is akin to the 2-D case treated in [1]. That the queue length trajectories under the average control $\bar{x}_i$ satisfy the initial and final conditions is easy to verify. Further, the trajectory moves along a straight line, and therefore stays entirely with $\mathcal{R}_{\mathcal{I}}$, due to the convexity of the region. Finally, due to the convexity of $\Lambda^*(\cdot)$, we have

$$\begin{aligned}\int_{-T_1}^{-T_2}\left[\sum_{i=1}^{N}\Lambda^*(x_i(t))\right]dt \geq\\ (T_1-T_2)\left[\sum_{i=1}^{N}\Lambda^*\left(\frac{1}{T_1-T_2}\int_{-T_1}^{-T_2}x_i(t)dt\right)\right]=\\ (T_1-T_2)\left[\sum_{i=1}^{N}\Lambda^*\left(\bar{x}_i\right)\right].\end{aligned}$$

This implies that the average control trajectory is not more costly than the original control trajectory. □

Using Lemma 2, we next compute the exponents corresponding to overflow trajectories that stay entirely within a particular region $\mathcal{R}_{\mathcal{I}}$. Later, we will show that overflow trajectories that traverse more than one region cannot have a strictly smaller exponent than trajectories that stay within exactly one of the regions. This will give us the result we want.

Consider an overflow trajectory that lies entirely within $\mathcal{R}_{\mathcal{I}_j}$, where $|\mathcal{I}_j|=j$ for some $1\leq j<N$. In this case, the j queues in the index set $\mathcal{I}_j$ reach overflow, while the other $N-j$ queues are strictly smaller and, hence, receive no service. Due to the symmetry of arrivals, we can compute the exponent assuming that $\mathcal{I}_j=\{1,\dots,j\}$, i.e., the first j queues overflow. Lemma 2 implies that the optimal empirical rates can be restricted to constant values[5] $x_i,\ i=1,\dots,N$ for this particular overflow event. Let $a=1/T$ denote the rate at which the first j queues overflow. Since each queue $k\in\{1,\dots,j\}$ overflows at rate a, the empirical input rate x_k must be of the form $x_k=a+\phi_k$, where $\phi_k\geq 0$ can be thought of as the rate at which queue k receives service in the overflow interval. Since the first j queues receive all the service, we have $\sum_{k=1}^{j}\phi_k=1$. Next, for $l>j$, we need $x_l\leq a$, since these queues are never the longest, and hence get no service.

The optimization in (15) takes the following form when the first j queues reach overflow:

$$\inf_{a>0}\frac{1}{a}\inf_{\substack{\phi_k\geq 0;\sum_{k=1}^{j}\phi_k=1\\ x_l\leq a,\ \forall\ l>j}}\sum_{k=1}^{j}\Lambda^*(a+\phi_k)+\sum_{l=j+1}^{N}\Lambda^*(x_l). \tag{16}$$

Let us now perform the inner minimization in (16). It is obvious that the minimization over $\phi_k,\ k\leq j$ and $x_l,\ l>j$, can be performed independently. Due to convexity of the rate function, we have

$$\frac{1}{j}\sum_{k=1}^{j}\Lambda^*(a+\phi_k)\geq\Lambda^*(\frac{1}{j}\sum_{k=1}^{j}(a+\phi_k))=\Lambda^*(a+\frac{1}{j}).$$

Therefore, the optimal value of ϕ_ks is given by $\phi_k=1/j,\ k\leq j$. Next, consider optimizing over x_l for $l>j$. We distinguish two cases:

i) $a>\lambda$: In this case, it is optimal to choose $x_l=\lambda$ for each $l>j$, since $\Lambda^*(\lambda)=0$.

ii) $a\leq\lambda$: In this case, the constraint $x_l\leq a$ has to be active, since for $x<\lambda,\ \Lambda^*(x)$ is decreasing in x. Thus, we have $x_l=a$.

Putting the two cases together, we get from (16) the exponent E_j corresponding to exactly j queues overflowing, while the trajectory stays inside $\mathcal{R}_{\mathcal{I}_j}$.

$$E_j=\min(\chi_j,\xi_j) \tag{17}$$

with

$$\chi_j=\inf_{0<a\leq\lambda}\frac{1}{a}\left[j\Lambda^*(a+\frac{1}{j})+(N-j)\Lambda^*(a)\right],\ \text{and}$$

$$\xi_j=\inf_{a>\lambda}\frac{j}{a}\Lambda^*(a+\frac{1}{j}). \tag{18}$$

[5]For simplicity of notation, we henceforth use x_i in place of $\bar{x}_i$.

The above expression holds for $1 \leq j < N$. The exponent for all the N queues overflowing is simpler to obtain; it is given by

$$E_N = \inf_{a>0} \frac{N}{a} \Lambda^*(a + \frac{1}{N}) = N\Theta_N, \tag{19}$$

where the last equality follows by recalling (6). The optimal exponent considering the set of all overflow trajectories that stay inside *any one* of the regions $\mathcal{R}_{\mathcal{I}},\ \mathcal{I} \subseteq \{1, \ldots, N\}$ is obtained by minimizing E_j over $j = 1, \ldots, N$.

At this point, we are two steps away from obtaining the result. The first step involves showing that there is nothing further to be gained by considering paths that traverse more than one of the partitioning regions during the interval $(-T, 0)$. This would imply that the optimal exponent is given by $\min_{1 \leq j \leq N} E_j$. The second step involves showing that $\min_{1 \leq j \leq N} E_j = \min_{j \in \mathcal{I}^*} j\Theta_j$, where Θ_j is the system occupancy exponent of j parallel queues, defined in (10). The following two lemmas establish what is needed.

Lemma 3: For every queue overflow trajectory that traverses more than one of the regions $\mathcal{R}_{\mathcal{I}},\ \mathcal{I} \subset \{1, \ldots, N\}$, there exists an overflow trajectory that lies entirely within one of the regions, while achieving an exponent that is no larger.

Proof: We only rule out overflow trajectories that traverse two regions; similar arguments can be successively used for trajectories that visit more than two regions. Consider a trajectory that starts out in a region $\mathcal{R}_{\mathcal{I}}$ but reaches overflow in region $\mathcal{R}_{\mathcal{J}}$, while staying in one of the two regions at every instant in between. Note that the region $\mathcal{R}_{\mathcal{I}}$ is a convex set of dimension $N - |\mathcal{I}| + 1$.

There are three possibilities: $\mathcal{I} \subset \mathcal{J}$, $\mathcal{J} \subset \mathcal{I}$, or neither.

First, suppose $\mathcal{I} \subset \mathcal{J}$. Consider a trajectory that remains in $\mathcal{R}_{\mathcal{I}}$ during the interval $(-T, -T_1)$, and stays in $\mathcal{R}_{\mathcal{J}}$ during the interval $[-T_1, 0)$, until overflow at $t = 0$. Intuitively, the queues $q_i,\ i \in \mathcal{I}$ start out growing together. At time $-T_1$, the queues $q_i,\ i \in \mathcal{J} - \mathcal{I}$ 'catch up', and overflow occurs in all the queues in the index set $\mathcal{J}$.Fig. 5(a) depicts a 2-D example corresponding to this scenario, where the second queue catches up with the first queue, and the two queues reach overflow together.

Since constant empirical input rates are optimal inside each partition region (Lemma 2), the arbitrary trajectory in $\mathcal{R}_{\mathcal{I}}$ can be replaced at no further cost by a straight segment that has the same end points. To be more precise, for an arbitrarily small $\epsilon > 0$, the trajectory in $\mathcal{R}_{\mathcal{I}}$ during the interval $[-T+\epsilon, -T_1-\epsilon]$ can be replaced by a straight segment joining the corresponding end points. This segment lies entirely inside $\mathcal{R}_{\mathcal{I}}$, but is arbitrarily close to the region $\mathcal{R}_{\mathcal{J}}$. Next, due to continuity, the cost of this replaced segment as $\epsilon \downarrow 0$ is not lower than the optimal trajectory in $\mathcal{R}_{\mathcal{J}}$ with the same boundary conditions as the original trajectory. Finally, the part of the original trajectory from $t = -T_1$ until over flow at $t = 0$, can again be replaced by the optimal trajectory in $\mathcal{R}_{\mathcal{J}}$ with the corresponding end points. Thus, overall, the cost of the original trajectory is greater than or equal to that of the optimal trajectory in $\mathcal{R}_{\mathcal{J}}$ with the same boundary condition at $t = 0$.

Next, consider the case $\mathcal{I} \supset \mathcal{J}$. Intuitively, this case corresponds to the queues $q_i,\ i \in \mathcal{I}$, starting to grow together. At some time instant, the queues $q_i,\ i \in \mathcal{I} - \mathcal{J}$, start "losing out," and overflow occurs within $\mathcal{R}_{\mathcal{J}}$. Fig. 5(b) depicts a 2-D example, where both queues start out growing together, but the second queue loses out and only the first queue reaches overflow.

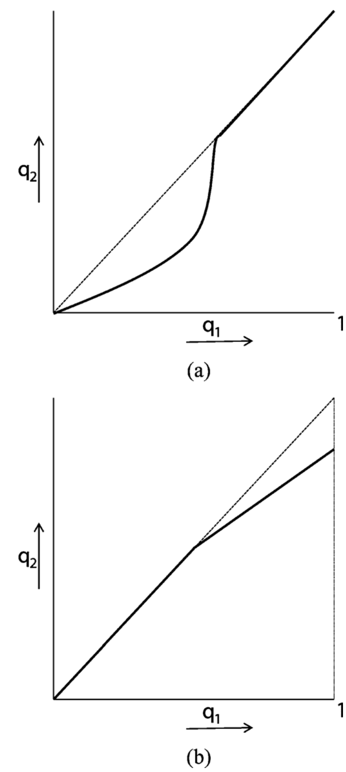


Fig. 5. Two-dimensional example to illustrate the trajectories considered in the proof of Lemma 3. (a) depicts the case where the second queue catches up with the first to overflow together. In scenario (b), the two queues start out growing together, but the second queue loses out and only the first queue reaches overflow.

The arbitrary trajectories in each of the regions can be replaced with an optimal segment in each of the regions, with the same boundary conditions at no added cost. The cost of this replaced trajectory is a convex combination of the optimal overflow trajectories in regions $\mathcal{R}_{\mathcal{J}}$ and $\mathcal{R}_{\mathcal{I}}$, and hence cannot be smaller than the smaller of the two costs.

Finally, consider the third case, i.e., $\mathcal{I} \nsubseteq \mathcal{J}$ and $\mathcal{J} \nsubseteq \mathcal{I}$. In this case, it can be shown that a trajectory that starts out in $\mathcal{R}_{\mathcal{I}}$ to reach overflow in $\mathcal{R}_{\mathcal{J}}$ has to traverse at least one more region. In particular, we will argue that such a trajectory must traverse $\mathcal{R}_{\mathcal{K}}$, for some $\mathcal{K} \supseteq \mathcal{I} \cup \mathcal{J}$. To see this, suppose on the contrary that $\mathbf{q}(-t_1) \in \mathcal{R}_{\mathcal{I}},\ \mathbf{q}(-t_2) \in \mathcal{R}_{\mathcal{J}}$, but there is no time instant in the interval $(-t_1, -t_2)$ when the system is in $\mathcal{R}_{\mathcal{K}}$. This would imply that the arrival trajectory $\mathbf{a}(t)$ contains a discontinuity in the interval $(-t_1, -t_2)$, since the service process is deterministically bounded. Recall that we are optimizing only over absolutely continuous arrival trajectories, and that discontinuities in $\mathbf{a}(t)$ correspond to an infinite cost, according to the LDP of the arrival process. Thus, when the system is driven by absolutely continuous arrival trajectories, the queue trajectory must traverse at least three regions. Such a trajectory can be shown to be suboptimal by considering the regions pairwise and using arguments similar to the other two cases. $\square$

Lemma 4: $\min_{1 \leq j \leq N} E_j = \min_{j \in \mathcal{I}^*} j\Theta_j$.

Proof: We first prove that $\chi_j \geq E_N$ for all $j < N$. First, using convexity of $\Lambda^*(\cdot)$, we can write

$$\frac{j}{N}\Lambda^*(a+\frac{1}{j}) + \frac{N-j}{N}\Lambda^*(a) \geq \Lambda^*\left(\frac{j}{N}(a+\frac{1}{j}) + \frac{N-j}{N}a\right) = \Lambda^*(a+\frac{1}{N}). \quad (20)$$

We now have

$$\begin{aligned}\chi_j &= \inf_{0<a\leq\lambda} \frac{1}{a}\left[j\Lambda^*(a+\frac{1}{j}) + (N-j)\Lambda^*(a)\right] \\ &\geq \inf_{a>0} \frac{1}{a}\left[j\Lambda^*(a+\frac{1}{j}) + (N-j)\Lambda^*(a)\right] \\ &\overset{(a)}{\geq} \inf_{a>0} \frac{N}{a}\Lambda^*(a+\frac{1}{N}) = E_N.\end{aligned}$$

The inequality (a) follows from (20). It is now clear that the χ_js are irrelevant, as they are always dominated by $E_N = N\Theta_N$. Taking $\xi_N = N\Theta_N$, we next write the following series of equalities that imply the lemma:

$$\begin{aligned}\min_{1\leq j\leq N} E_j &= \min(\xi_1, \ldots, \xi_{N-1}, N\Theta_N) \\ &= \min_{1\leq j\leq N} \xi_j \\ &= \min_{j\in\mathcal{I}^*} \xi_j \quad (21) \\ &= \min_{j\in\mathcal{I}^*} j\Theta_j. \quad (22)\end{aligned}$$

In the above, equality (22) follows from the definition of $\mathcal{I}^*$, and (21) is shown as follows. Using the fact that $\Lambda^*(a+\frac{1}{j})$ is convex in a, it can be shown that $\frac{1}{a}\Lambda^*(a+\frac{1}{j})$ is increasing for all $a > a_j^*$. Now, consider some $i \notin \mathcal{I}^*$, so that $a_i^* \leq \lambda$. The definition of ξ_i(18) involves taking the infimum over $a > \lambda$. Since $a_i^* < \lambda$, the infimum in (18) is attained at $a = \lambda$. In this case, it is easy to show that $\xi_N \leq \xi_i$. Indeed, for $i \notin \mathcal{I}^*$, we have

$$\begin{aligned}\xi_i &= \frac{i}{\lambda}\Lambda^*(\lambda+\frac{1}{i}) \\ &= \frac{i}{\lambda}\Lambda^*(\lambda+\frac{1}{i}) + \frac{N-i}{\lambda}\Lambda^*(\lambda) \\ &\overset{(b)}{\geq} \frac{N}{\lambda}\Lambda^*(\lambda+\frac{1}{N}) \\ &\geq \xi_N,\end{aligned}$$

where step (b) is due to convexity. Thus, we have shown (21). □

2) Proof Outline of Proposition 6: Let $\Phi_i[t] \in \{0,1\}$ denote the i.i.d. fair "coin tosses" that decide which queue to serve when both queues are occupied. If $\Phi_i[t] = 1$, then the second queue is served if occupied in slot t; if $\Phi_i[t] = 0$, the first queue is served if occupied. Define $\Phi[-T,t] = \sum_{\tau=-T}^{t} \Phi_i[\tau]$, and $\Phi^{(M)}(t) = \frac{1}{M}\Phi[-MT, \lfloor Mt \rfloor],\ t \in [-T, 0]$. The sequence $\Phi^{(M)}(t)$ satisfies an LDP with rate function

$$I(\Phi(t)) = \int_{-T}^{0} D(\phi(t)\|1/2)dt,$$

for absolutely continuous trajectories $\Phi(t)$, with $\Phi(-T) = 0$, and $\phi(t)$ is the almost everywhere derivative of $\Phi(t)$.

The dynamics of the fluid queue length processes under RS is given by

$$\begin{aligned}\dot{q}_1(t) &= x_1(t) - (1-\phi(t)) \\ \dot{q}_2(t) &= x_2(t) - \phi(t),\end{aligned}$$

whenever $q_1(t)$ and $q_2(t)$ are nonzero. If either $q_1(t) = 0$ or $q_2(t) = 0$, then

$$\dot{q}_1(t) + \dot{q}_2(t) = x_1(t) + x_2(t) - 1.$$

Here, $x_1(t)$ and $x_2(t)$ are the empirical rates of the input processes.

Using a result analogous to Lemma 2, we can show that constant empirical rates for the inputs as well as the coin tosses is optimal, within each of the regions (i) $q_1(t) > 0,\ q_2(t) > 0$ (ii) $q_1(t) > 0,\ q_2(t) = 0$, and (iii) $q_1(t) = 0,\ q_2(t) > 0$. For a given empirical rate ϕ, the problem can be mapped to an instance of GPS, as treated in [1]. The result follows by applying the GPS exponent results to our symmetric case, and noting that the rate function corresponding to the fair coin tosses is given by $D(\cdot\|1/2)$.

3) Proof of Theorem 2: Let us first prove part (i), which is quite straightforward. Given $\delta > 0$, suppose that we wish to make the fraction of slots in which queue length information is required less than δ. Since the hybrid policy is work-conserving for every K, the steady-state probability of the largest queue exceeding K approaches zero as K becomes large. In other words, we can choose a K_δ such that for any $K > K_\delta$, the probability of the longest queue exceeding K is less than δ. It is now clear that a hybrid policy with $K > K_\delta$ will achieve what we want, since the hybrid policy requires queue length information only in slots when the longest queue exceeds K.

We now proceed to show part (ii) of the theorem. For any fixed parameter K of the hybrid policy, we will show that the overflow exponent remains the same as that of the LQF policy. We first prove an elementary lemma, which asserts that given two systems with different initial queue occupancies, the LQF policy does not allow the queue evolution trajectories to "diverge," when the two systems are fed by the same input process.

Definition 1: For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^N$, define $d(\mathbf{x},\mathbf{y}) = \max_{i=1,\ldots,N} |x_i - y_i|$.

Lemma 5: Consider two fictitious systems U and V, in which the initial queue lengths (at time zero) are given by $Q_i^{(U)}[0],\ i = 1,\ldots,N$ and $Q_i^{(V)}[0],\ i = 1,\ldots,N$, respectively. Let $\Delta = d(\mathbf{Q}^{(U)}[0], \mathbf{Q}^{(V)}[0])$. Suppose that

a) The same input sample path $\mathbf{A}[t],\ t = 1,\ldots,T_0$ feeds both systems for T_0 time slots, and that
b) LQF scheduling is performed (with the same tie breaking rule) on both the systems for $t = 1,\ldots,T_0$.

Then, for any input sample path and T_0, we have $d(\mathbf{Q}^{(U)}[T_0], \mathbf{Q}^{(V)}[T_0]) \leq \Delta$.

Proof: The queue lengths in the two systems after the arrivals during the first time slot are given by $B_i^{(U)}[1] = Q_i^{(U)}[0] + A_i[1], i = 1,\ldots,N$, and $B_i^{(V)}[1] = Q_i^{(V)}[0] + A_i[1], i = 1,\ldots,N$. At the end of the first time slot, LQF scheduling is performed based on these queue lengths. We will consider the following exhaustive possibilities and show that $d(\mathbf{Q}^{(U)}[1], \mathbf{Q}^{(V)}[1]) \leq \Delta$.

i) If the LQF policy chooses the same queue to serve in both system, it is clear that $d(\mathbf{Q}^{(U)}[1], \mathbf{Q}^{(V)}[1]) = \Delta$.

ii) Suppose the LQF policy chooses queue u in system U and queue v in system V, with $u \neq v$. This implies that $B_u^{(U)}[1] \geq B_v^{(U)}[1]$ and $B_u^{(V)}[1] \leq B_v^{(V)}[1]$. The following subcases arise:

(iia) $B_u^{(U)}[1] > B_u^{(V)}[1]$ and $B_v^{(V)}[1] > B_v^{(U)}[1]$. In this case, after the LQF policy finishes service, $|Q_i^{(U)}[1] - Q_i^{(V)}[1]| = |Q_i^{(U)}[0] - Q_i^{(V)}[0]| - 1$ for $i = u, v$, and $|Q_i^{(U)}[1] - Q_i^{(V)}[1]| = |Q_i^{(U)}[0] - Q_i^{(V)}[0]|$ $i \neq u, v$. Thus, in this case, $d(\mathbf{Q}^{(U)}[1], \mathbf{Q}^{(V)}[1]) \leq \Delta$.

(iib) One of the inequalities in (iia) fails to hold. Suppose $B_u^{(U)}[1] \leq B_u^{(V)}[1]$. This implies that $B_v^{(U)}[1] \leq_a B_u^{(U)}[1] \leq_b B_u^{(V)}[1] \leq_c B_v^{(V)}[1]$. We can assume that at least one of the inequalities a and c is strict, for if not, there is a tie in both systems between queues u and v, and this case is covered under (i). It is then clear that $|Q_v^{(U)}[0] - Q_v^{(V)}[0]| > |Q_u^{(U)}[0] - Q_u^{(V)}[0]|$. Since only one packet can depart during a time slot, we have $|Q_u^{(U)}[1] - Q_u^{(V)}[1]| = |Q_u^{(U)}[0] - Q_u^{(V)}[0]| + 1 \leq |Q_v^{(U)}[0] - Q_v^{(V)}[0]| \leq \Delta$, and $|Q_v^{(U)}[1] - Q_v^{(V)}[1]| = |Q_v^{(U)}[0] - Q_v^{(V)}[0]| - 1$. Thus, $d(\mathbf{Q}^{(U)}[1], \mathbf{Q}^{(V)}[1]) \leq \Delta$ in this case too.

Iterating over time slots, it can be shown that $d(\mathbf{Q}^{(U)}[t], \mathbf{Q}^{(V)}[t]) \leq \Delta$, for all $t \geq 1$. □

Let us now show that the overflow exponent under hybrid scheduling is greater than or equal to E_N^{LQF}. We prove this by showing that for every input sample path that leads to a buffer overflow under hybrid scheduling, the LQF policy also gets close to overflow. Specifically, let $\mathbf{A}[t],\ t = 1, \ldots, T$, be an input sample path which leads to a buffer overflow at time T under hybrid scheduling. Thus, $Q_i^{(H)}[T] \geq M$, for some $i \leq N$. (We use the superscript H to denote queue lengths under hybrid scheduling, and L for LQF scheduling). Let $\tau \leq T$ denote the last time that all the queues were less than or equal to K. Thus, $Q_j^{(H)}[\tau] \leq K, j = 1, \ldots, N$. Now, since both hybrid scheduling and LQF scheduling are work-conserving, the total number of packets in the system is conserved. Thus, if the same input sample path were to feed a system with LQF scheduling in each slot, we would have $\sum_j Q_j^{(L)}[\tau] = \sum_j Q_j^{(H)}[\tau] < NK$, from which it is immediate that

$$d(\mathbf{Q}^{(H)}[\tau], \mathbf{Q}^{(L)}[\tau]) < NK. \tag{23}$$

Observe that by the definition of τ, the hybrid policy actually performs LQF scheduling during the time slots $\tau + 1, \ldots, T$. Thus, we have two systems which start with different initial queue lengths $\mathbf{Q}^{(H)}[\tau]$ and $\mathbf{Q}^{(L)}[\tau]$. However, they are both fed by the same input sample path and are served according to the LQF policy for $t > \tau$. Lemma 5 now applies, and we can conclude that $d(\mathbf{Q}^{(H)}[T], \mathbf{Q}^{(L)}[T]) \leq d(\mathbf{Q}^{(H)}[\tau], \mathbf{Q}^{(L)}[\tau])$. When combined with (23), this yields $d(\mathbf{Q}^{(H)}[T], \mathbf{Q}^{(L)}[T]) < NK$. Thus, $Q_i^{(L)}[T] \geq M - NK$, whenever $Q_i^{(H)}[T] \geq M$. This shows that for every input sample path that leads to an overflow under Hybrid scheduling, the LQF policy is also close to overflow.

Since this is true for every overflow sample path, we have the steady-state relation

$$\mathbb{P}\left\{\max_i Q_i^{(H)} \geq M\right\} \leq \mathbb{P}\left\{\max_i Q_i^{(L)} \geq M - NK\right\},$$

from which it follows that $E_N^H \geq E_N^{\text{LQF}}$. Next, in order to show that $E_N^H \leq E_N^{\text{LQF}}$, we can argue as above that every input sample path that leads to overflow under LQF scheduling, also leads "close" to an overflow under hybrid scheduling. We have shown that for a fixed K, the hybrid scheduling policy has overflow exponent equal to E_N^{LQF}. It is not difficult to see that if K increases sublinearly in M, i.e., $K = o(M)$, the exponent would still remain the same. This implies that by scaling K sublinearly in the buffer size M, the rate of queue length information can be sent to zero, while still achieving the exponent corresponding to LQF scheduling. □

ACKNOWLEDGMENT

The authors are grateful to Prof. J. Tsitsiklis for his valuable comments and suggestions, and to the anonymous reviewers for taking exceptional care in reviewing the paper.

REFERENCES

[1] D. Bertsimas, I. Paschalidis, and J. Tsitsiklis, "Asymptotic buffer overflow probabilities in multiclass multiplexers: An optimal control approach," *IEEE Trans. Autom. Control*, vol. 43, no. 3, pp. 315–335, Mar. 1998.

[2] A. A. Borovkov, "Boundary-value problems for random walks and large deviations in function spaces," *Theory Probab. Appl.*, vol. 12, pp. 575–575, 1967.

[3] S. Borst, R. Núñez-Queija, and B. Zwart, "Sojourn time asymptotics in processor-sharing queues," *Queu. Syst.*, vol. 53, no. 1, pp. 31–51, 2006.

[4] A. Dembo and T. Zajic, "Large deviations: from empirical mean and measure to partial sums process," *Stochast. Process. Appl.*, vol. 57, no. 2, pp. 191–224, 1995.

[5] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. New York, NY, USA: Springer-Verlag, 2009.

[6] A. Ganesh, N. O'Connell, and D. Wischik, *Big Queues*. New York, NY, USA: Springer-Verlag, 2004.

[7] P. Glynn and W. Whitt, "Ordinary CLT and WLLN versions of $l = \lambda w$," *Math. Oper. Res.*, pp. 674–692, 1988.

[8] K. Jagannathan and E. Modiano, "The impact of queue length information on buffer overflow in parallel queues," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput.*, 2009, pp. 1103–1110.

[9] A. Mogul'skii, "Large deviations for trajectories of multi-dimensional random walks," *Theory Probab. Appl.*, vol. 21, pp. 300–315, 1976.

[10] A. Mogulskii, "Large deviations for processes with independent increments," *Ann. Probab.*, pp. 202–215, 1993.

[11] M. Neely, E. Modiano, and C. Rohrs, "Power and server allocation in a multi-beam satellite with time varying channels," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2002, vol. 3, pp. 1451–1460.

[12] S. Shakkottai, "Effective capacity and QoS for wireless scheduling," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 749–761, Apr. 2008.

[13] A. Stolyar, "Control of end-to-end delay tails in a multiclass network: LWDF discipline optimality," *Ann. Appl. Probab.*, vol. 13, pp. 1151–1206, 2003.

[14] A. Stolyar and K. Ramanan, "Largest weighted delay first scheduling: Large deviations and optimality," *Ann. Appl. Probab.*, vol. 11, no. 1, pp. 1–48, 2001.

[15] V. G. Subramanian, "Large deviations of max-weight scheduling policies on convex rate regions," *Math. Oper. Res.*, vol. 35, no. 4, pp. 881–881, 2010.

[16] 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.

[17] S. R. S. Varadhan, "Asymptotic probabilities and differential equations," *Commun. Pure Appl. Math.*, vol. 19, no. 3, pp. 261–286, 1966.

[18] V. Venkataramanan and X. Lin, "On the large-deviations optimality of scheduling policies minimizing the drift of a Lyapunov function," presented at the 47th Annu. Allert. Conf. Commun., Control, Comput., Monticello, IL, USA, 2009.

[19] V. Venkataramanan and X. Lin, "On the queue-overflow probability of wireless systems: A new approach combining large deviations with Lyapunov functions," *IEEE Trans. Inf. Theory*, to be published.

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

[21] V. Venkataramanan, X. Lin, S. Shakkottai, and L. Ying, "On scheduling for minimizing end-to-end buffer usage over multihop wireless networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

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

Krishna Jagannathan (S'06–M'12) is an assistant professor in the Department of Electrical Engineering, Indian Institute of Technology (IIT) Madras, India. He obtained his B. Tech. in Electrical Engineering from IIT Madras in 2004, and the S.M. and Ph.D. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT) in 2006 and 2010 respectively. During 2010–2011, he was a visiting post-doctoral scholar in Computing and Mathematical Sciences at Caltech, and an off-campus post-doctoral fellow at MIT. He worked as a consultant at the Mathematical Sciences Research Center, Bell Laboratories, Murray Hill, NJ in 2005, an engineering intern at Qualcomm, Campbell, CA in 2007, and a summer research fellow at the Indian Institute of Science (IISc), Bangalore, India in 2003. His research interests lie in communication networks, queuing theory, and network control.

Eytan Modiano (S'90–M'93–SM'00–F'12) received his B.S. degree in Electrical Engineering and Computer Science from the University of Connecticut at Storrs in 1986 and his M.S. and PhD degrees, both in Electrical Engineering, from the University of Maryland, College Park, MD, in 1989 and 1992 respectively. He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Post Doctoral Fellow during 1992–1993. Between 1993 and 1999 he was with MIT Lincoln Laboratory where he was a project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. Since 1999 he has been on the faculty at MIT; where he is a Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks. He is the co-recipient of the Sigmetrics 2006 Best paper award for the paper "Maximizing Throughput in Wireless Networks via Gossiping," and the Wiopt 2005 best student paper award for the paper "Minimum Energy Transmission Scheduling Subject to Deadline Constraints."

He is currently an Associate Editor for IEEE/ACM TRANSACTIONS ON NETWORKING. He had served as Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY, and as guest editor for IEEE JSAC special issue on WDM network architectures; the *Computer Networks Journal* special issue on Broadband Internet Access; the *Journal of Communications and Networks* special issue on Wireless Ad-Hoc Networks; and for the IEEE JOURNAL OF LIGHTWAVE TECHNOLOGY special issue on Optical Networks. He was the Technical Program co-chair for IEEE Wiopt 2006, IEEE Infocom 2007, and ACM MobiHoc 2007. He is a Fellow of the IEEE and Associate Fellow of the AIAA.