The Annals of Applied Probability
2014, Vol. 24, No. 5, 2091–2142
DOI: 10.1214/13-AAP973

QUEUING WITH FUTURE INFORMATION

BY JOEL SPENCER, MADHU SUDAN AND KUANG XU[1]

New York University, Microsoft Research New England and Massachusetts Institute of Technology

We study an admissions control problem, where a queue with service rate $1-p$ receives incoming jobs at rate $\lambda \in (1-p, 1)$, and the decision maker is allowed to redirect away jobs up to a rate of p, with the objective of minimizing the time-average queue length.

We show that the amount of *information about the future* has a significant impact on system performance, in the heavy-traffic regime. When the future is unknown, the optimal average queue length diverges at rate $\sim \log_{1/(1-p)} \frac{1}{1-\lambda}$, as $\lambda \to 1$. In sharp contrast, when all future arrival and service times are revealed beforehand, the optimal average queue length converges to a finite constant, $(1-p)/p$, as $\lambda \to 1$. We further show that the finite limit of $(1-p)/p$ can be achieved using only a *finite* lookahead window starting from the current time frame, whose length scales as $\mathcal{O}(\log \frac{1}{1-\lambda})$, as $\lambda \to 1$. This leads to the conjecture of an interesting duality between queuing delay and the amount of information about the future.

1. Introduction.

1.1. *Variable, but predictable.* The notion of *queues* has been used extensively as a powerful abstraction in studying dynamic resource allocation systems, where one aims to match *demands* that arrive over time with available *resources*, and a queue is used to store currently unprocessed demands. Two important ingredients often make the design and analysis of a queueing system difficult: the demands and resources can be both *variable* and *unpredictable*. *Variability* refers to the fact that the arrivals of demands or the availability of resources can be highly volatile and nonuniformly distributed across the time horizon. *Unpredictability* means that such nonuniformity "tomorrow" is unknown to the decision maker "today," and she is obliged to make allocation decisions only based on the state of the system at the moment, and some statistical estimates of the future.

While the world will remain volatile as we know it, in many cases, the amount of unpredictability about the future may be reduced thanks to *forecasting* technologies and the increasing accessibility of data. For instance:

Received January 2013; revised April 2013.

[1]Supported in part by a research internship at Microsoft Research New England and by NSF Grants CMMI-0856063 and CMMI-1234062.

MSC2010 subject classifications. 60K25, 60K30, 68M20, 90B36.

Key words and phrases. Future information, queuing theory, admissions control, resource pooling, random walk, online, offline, heavy-traffic asymptotics.

(1) advance booking in the hotel and textile industries allows for accurate forecasting of demands ahead of time [9];

(2) the availability of monitoring data enables traffic controllers to predict the traffic pattern around potential bottlenecks [18];

(3) advance scheduling for elective surgeries could inform care providers several weeks before the intended appointment [12].

In all of these examples, future demands remain *exogenous* and variable, yet the decision maker is revealed with (some of) their realizations.

Is there significant performance gain to be harnessed by "*looking into the future*?" In this paper we provide a largely affirmative answer, in the context of a class of admissions control problems.

1.2. *Admissions control viewed as resource allocation.* We begin by informally describing our problem. Consider a single queue equipped with a server that runs at rate $1 - p$ jobs per unit time, where p is a fixed constant in $(0, 1)$, as depicted in Figure 1. The queue receives a stream of incoming jobs, arriving at rate $\lambda \in (0, 1)$. If $\lambda > 1 - p$, the arrival rate is greater than the server's processing rate, and some form of *admissions control* is necessary in order to keep the system stable. In particular, upon its arrival to the system, a job will either be *admitted* to the queue, or *redirected*. In the latter case, the job does not join the queue, and, from the perspective of the queue, disappears from the system entirely. The goal of the decision maker is to minimize the average delay experienced by the admitted jobs, while obeying the constraint that the average rate at which jobs are redirected *does not exceeded* p.[2]

One can think of our problem as that of *resource allocation*, where a decision maker tries to match incoming demands with two types of processing resources: a *slow local resource* that corresponds to the server and a *fast external resource* that can process any job redirected to it almost instantaneously. Both types of resources are *constrained*, in the sense that their capacities ($1 - p$ and p, resp.) cannot change over time, by physical or contractual predispositions. The processing time of a job at the fast resource is *negligible compared to that at the slow*

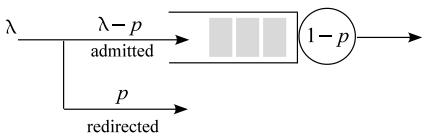

FIG. 1. *An illustration of the admissions control problem, with a constraint on the a rate of redirection.*

[2]Note that as $\lambda \to 1$, the minimum rate of admitted jobs, $\lambda - p$, approaches the server's capacity $1 - p$, and hence we will refer to the system's behavior when $\lambda \to 1$ as the *heavy-traffic regime*.

resource, as long as the rate of redirection to the fast resource stays below p in the long run. Under this interpretation, minimizing the average delay across *all* jobs is equivalent to minimizing the average delay across just the *admitted* jobs, since the jobs redirected to the fast resource can be thought of being processed immediately and experiencing no delay at all.

For a more concrete example, consider a web service company that enters a long-term contract with an external cloud computing provider for a fixed amount of computation resources (e.g., virtual machine instance time) over the contract period.[3] During the contract period, any incoming request can be either served by the in-house server (slow resource), or be redirected to the cloud (fast resource), and in the latter case, the job does not experience congestion delay since the scalability of cloud allows for multiple VM instance to be running in parallel (and potentially on different physical machines). The decision maker's constraint is that the total amount of redirected jobs to the cloud must stay below the amount prescribed by the contract, which, in our case, translates into a maximum redirection rate over the contract period. Similar scenarios can also arise in other domains, where the slow versus fast resources could, for instance, take on the forms of:

(1) an in-house manufacturing facility versus an external contractor;
(2) a slow toll booth on the freeway versus a special lane that lets a car pass without paying the toll;
(3) hospital bed resources within a single department versus a cross-departmental central bed pool.

In a recent work [20], a mathematical model was proposed to study the benefits of resource pooling in large scale queueing systems, which is also closely connected to our problem. They consider a multi-server system where a fraction $1 - p$ of a total of N units of processing resources (e.g., CPUs) is distributed among a set of N local servers, each running at rate $1 - p$, while the remaining fraction of p is being allocated in a centralized fashion, in the form of a central server that operates at rate pN (Figure 2). It is not difficult to see, when N is large, the central server operates at a significantly faster speed than the local servers, so that a job processed at the central server experiences little or no delay. In fact, the admissions control problem studied in this paper is essentially the problem faced by one of the local servers, in the regime where N is large (Figure 3). This connection is explored in greater detail in Appendix B, where we discuss what the implications of our results in context of resource pooling systems.

[3] *Example*. As of September 2012, Microsoft's Windows Azure cloud services offer a 6-month contract for $71.99 per month, where the client is entitled for up to 750 hours of virtual machine (VM) instance time each month, and any additional usage would be charged at a 25% higher rate. Due to the large scale of the Azure data warehouses, the speed of any single VM instance can be treated as roughly constant and independent of the total number of instances that the client is running concurrently.

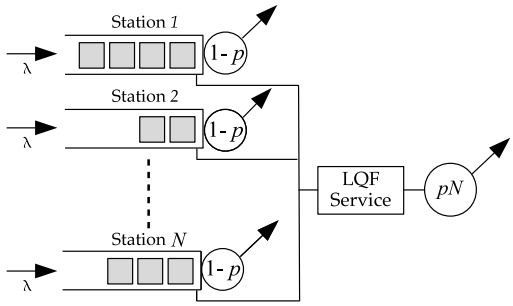

FIG. 2. *Illustration of a model for resource pooling with distributed and centralized resources* [20].

1.3. *Overview of main contributions*. We preview some of the main results in this section. The formal statements will be given in Section 3.

1.3.1. *Summary of the problem*. We consider a continuous-time admissions control problem, depicted in Figure 1. The problem is characterized by three parameters: λ, p and w:

(1) Jobs arrive to the system at a rate of λ jobs per unit time, with $\lambda \in (0, 1)$. The server operates at a rate of $1 - p$ jobs per unit time, with $p \in (0, 1)$.

(2) The decision maker is allowed to decide whether an arriving job is admitted to the queue, or redirected away, with the goal of minimizing the time-average queue length,[4] and subject to the constraint that the time-average rate of redirection does not exceed p jobs per unit time.

(3) The decision maker has access to *information about the future*, which takes the form of a *lookahead window* of length $w \in \mathbb{R}_+$. In particular, at any time t, the times of arrivals and service availability within the interval $[t, t + w]$ are revealed to the decision maker. We will consider the following cases of w:

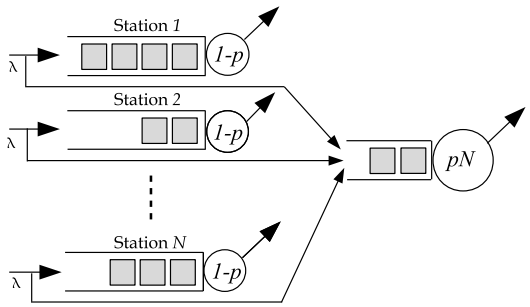

FIG. 3. *Resource pooling using a central queue.*

[4]By Little's law, the average queue length is essentially the same as average delay, up to a constant factor; see Section 2.5.

(a) $w = 0$, the *online problem*, where no future information is available.
(b) $w = \infty$, the *offline problem*, where entire the future has been revealed.
(c) $0 < w < \infty$, where future is revealed only up to a finite lookahead window.

Throughout, we will fix $p \in (0, 1)$, and be primarily interested in the system's behavior in the *heavy-traffic regime* of $\lambda \to 1$.

1.3.2. *Overview of main results.* Our main contribution is to demonstrate that the performance of a redirection policy is highly sensitive to the amount of future information available, measured by the value of w.

Fix $p \in (0, 1)$, and let the arrival and service processes be Poisson. For the online problem ($w = 0$), we show the optimal time-average queue length, C_0^{opt}, approaches infinity in the heavy-traffic regime, at the rate

$$C_0^{\text{opt}} \sim \log_{1/(1-p)} \frac{1}{1-\lambda} \qquad \text{as } \lambda \to 1.$$

In sharp contrast, the optimal average queue length among offline policies ($w = \infty$), C_∞^{opt}, converges to a *constant*,

$$C_\infty^{\text{opt}} \to \frac{1-p}{p} \qquad \text{as } \lambda \to 1$$

and this limit is achieved by a so-called no-job-left-behind policy. Figure 4 illustrates this difference in delay performance for a particular value of p.

Finally, we show that the no-job-left-behind policy for the offline problem can be modified, so that the *same* optimal heavy-traffic limit of $\frac{1-p}{p}$ is achieved even

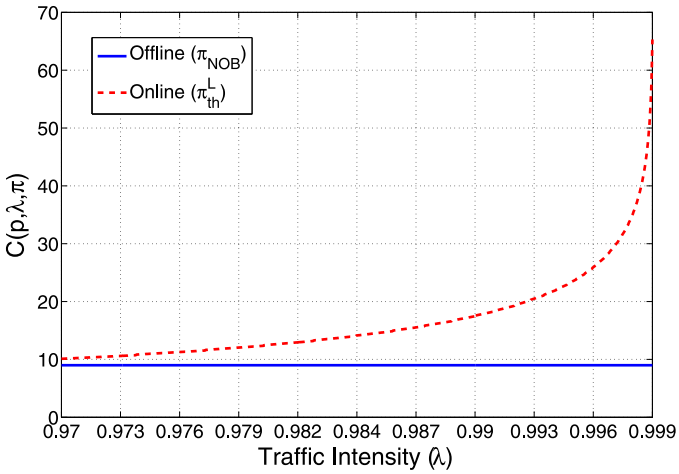

FIG. 4. *Comparison of optimal heavy-traffic delay scaling between online and offline policies, with* $p = 0.1$ *and* $\lambda \to 1$. *The value* $C(p, \lambda, \pi)$ *is the resulting average queue length as a function of* p, λ *and a policy* π.

with a *finite* lookahead window, $w(\lambda)$, where

$$w(\lambda) = \mathcal{O}\left(\log \frac{1}{1-\lambda}\right) \qquad \text{as } \lambda \to 1.$$

This is of practical importance because in any realistic application, only a finite amount of future information can be obtained.

On the methodological end, we use a sample path-based framework to analyze the performance of the offline and finite lookahead policies, borrowing tools from renewal theory and the theory of random walks. We believe that our techniques could be substantially generalized to incorporate general arrival and service processes, diffusion approximations as well as observational noises. See Section 8 for a more elaborate discussion.

1.4. *Related work.* There is an extensive body of work devoted to various Markov (or *online*) admissions control problems; the reader is referred to the survey of [19] and references therein. Typically, the problem is formulated as an instance of a Markov decision problem (MDP), where the decision maker, by admitting or rejecting incoming jobs, seeks to maximize a long-term average objective consisting of rewards (e.g., throughput) minus costs (e.g., waiting time experienced by a customer). The case where the maximization is performed subject to a constraint on some average cost has also been studied, and it has been shown, for a family of reward and cost functions, that an optimal policy assumes a "threshold-like" form, where the decision maker redirects the next job only if the current queue length is great or equal to L, with possible randomization if at level $L-1$, and always admits the job if below $L-1$; cf. [5]. Indeed, our problem, where one tries to minimize average queue length (delay) subject to a lower-bound on the throughput (i.e., a maximum redirection rate), can be shown to belong to this category, and the online heavy-traffic scaling result is a straightforward extension following the MDP framework, albeit dealing with technicalities in extending the threshold characterization to an infinite state space, since we are interested in the regime of $\lambda \to 1$.

However, the resource allocation interpretation of our admissions control problem as that of matching jobs with fast and slow resources, and, in particular, its connections to resource pooling in the many-server limit, seems to be largely unexplored. The difference in motivation perhaps explains why the optimal online heavy-traffic delay scaling of $\log_{1/(1-p)} \frac{1}{1-\lambda}$ that emerges by fixing p and taking $\lambda \to 1$ has not appeared in the literature, to the best our knowledge.

There is also an extensive literature on *competitive analysis*, which focuses on the *worst-case* performance of an online algorithms compared to that of an optimal offline version (i.e., knowing the entire input sequence). The reader is referred to [6] for a comprehensive survey, and the references therein on packing-type problems, such as load balancing and machine scheduling [3], and call admission and routing [2], which are more related to our problem. While our optimality result for

the policy with a finite lookahead window is stated in terms of the *average* performance given stochastic inputs, we believe that the analysis can be extended to yield worst-case competitive ratios under certain input regularity conditions.

In sharp contrast to our knowledge of the online problems, significantly less is known for settings in which information about the future is taken into consideration. In [17], the author considers a variant of the flow control problem where the decision maker knows the job size of the arriving customer, as well as the arrival and time and job size of the next customer, with the goal of maximizing certain discounted or average reward. A characterization of an optimal stationary policy is derived under a standard semi-Markov decision problem framework, since the lookahead is limited to the next arriving job. In [7], the authors consider a scheduling problem with one server and M parallel queues, motivated by applications in satellite systems where the link qualities between the server and the queues vary over time. The authors compare the throughput performance between several online policies with that of an offline policy, which has access to all future instances of link qualities. However, the offline policy takes the form of a Viterbi-like dynamic program, which, while being throughput-optimal by definition, provides limited qualitative insight.

One challenge that arises as one tries to move beyond the online setting is that policies with lookahead typically do not admit a clean Markov description, and hence common techniques for analyzing Markov decision problems do not easily apply. To circumvent the obstacle, we will first relax our problem to be fully offline, which turns out to be surprisingly amenable to analysis. We then use the insights from the optimal offline policy to construct an optimal policy with a finite lookahead window, in a rather straightforward manner.

In other application domains, the idea of exploiting future information or predictions to improve decision making has been explored. Advance reservations (a form of future information) have been studied in lossy networks [8, 14] and, more recently, in revenue management [13]. Using simulations, [12] demonstrates that the use of a one-week and two-week advance scheduling window for elective surgeries can improve the efficiency at the associated intensive care unit (ICU). The benefits of advanced booking program for supply chains have been shown in [9] in the form of reduced demand uncertainties. While similar in spirit, the motivations and dynamics in these models are very different from ours.

Finally, our formulation of the slow an fast resources had been in part inspired by the literature of resource pooling systems, where one improves overall system performance by (partially) sharing individual resources in collective manner. The connection of our problem to a specific multi-server model proposed by [20] is discussed in Appendix B. For the general topic of resource pooling, interested readers are referred to [4, 11, 15, 16] and the references therein.

1.5. *Organization of the paper.* The rest of the paper is organized as follows. The mathematical model for our problem is described in Section 2. Section 3 contains the statements of our main results, and introduces the no-job-leftb-behind

policy (π_{NOB}), which will be a central object of study for this paper. Section 4 presents two alternative descriptions of the no-job-left-behind policy that have important structural, as well as algorithmic, implications. Sections 5–7 are devoted to the proofs for the results concerning the online, offline and finite-lookahead policies, respectively. Finally, Section 8 contains some concluding remarks and future directions.

2. Model and setup.

2.1. *Notation*. We will denote by $\mathbb{N}$, $\mathbb{Z}_+$ and $\mathbb{R}_+$, the set of natural numbers, nonnegative integers and nonnegative reals, respectively. Let $f, g : \mathbb{R}_+ \to \mathbb{R}_+$ be two functions. We will use the following asymptotic notation throughout: $f(x) \lesssim g(x)$ if $\lim_{x\to 1} \frac{f(x)}{g(x)} \le 1$, $f(x) \gtrsim g(x)$ if $\lim_{x\to 1} \frac{f(x)}{g(x)} \ge 1$; $f(x) \ll g(x)$ if $\lim_{x\to 1} \frac{f(x)}{g(x)} = 0$ and $f(x) \gg g(x)$ if $\lim_{x\to 1} \frac{f(x)}{g(x)} = \infty$.

2.2. *System dynamics*. An illustration of the system setup is given in Figure 1. The system consists of a single-server queue running in continuous time ($t \in \mathbb{R}_+$), with an unbounded buffer that stores all unprocessed jobs. The queue is assumed to be empty at $t = 0$.

Jobs arrive to the system according to a Poisson process with rate λ, $\lambda \in (0, 1)$, so that the intervals between two adjacent arrivals are independent and exponentially distributed with mean $\frac{1}{\lambda}$. We will denote by $\{A(t) : t \in \mathbb{R}_+\}$ the cumulative arrival process, where $A(t) \in \mathbb{Z}_+$ is the total number of arrivals to the system by time t.

The processing of jobs by the server is modeled by a Poisson process of rate $1 - p$. When the service process receives a jump at time t, we say that a service token is generated. If the queue is not empty at time t, exactly one job "consumes" the service token and leaves the system immediately. Otherwise, the service token is "wasted" and has no impact on the future evolution of the system.[5] We will

[5]When the queue is nonempty, the generation of a token can be interpreted as the completion of a previous job, upon which the server is ready to fetch the next job. The time between two consecutive tokens corresponds to the service time. The waste of a token can be interpreted as the server starting to serve a "dummy job." Roughly speaking, the service token formulation, compared to that of a constant speed server processing jobs with exponentially distributed sizes, provides a performance upper-bound due to the inefficiency caused by dummy jobs, but has very similar performance in the heavy-traffic regime, in which the tokens are almost never wasted. Using such a point process to model services is not new, and the reader is referred to [20] and the references therein.

It is, however, important to note a key assumption implicit in the service token formulation: the processing times are intrinsic to the server, and *independent* of the job being processed. For instance, the sequence of service times will not depend on the order in which the jobs in the queue are served, so long as the server remains busy throughout the period. This distinction is of little relevance for an $M/M/1$ queue, but can be important in our case, where the redirection decisions may depend on the future. See discussion in Section 8.

denote by $\{S(t):t\in\mathbb{R}_+\}$ the cumulative token generation process, where $S(t)\in\mathbb{Z}_+$ is the total number of service tokens generated by time t.

When $\lambda>1-p$, in order to maintain the stability of the queue, a decision maker has the option of "redirecting" a job *at the moment of its arrival*. Once redirected, a job effectively "disappears," and for this reason, we will use the word *deletion* as a synonymous term for redirection throughout the rest of the paper, because it is more intuitive to think of deleting a job in our subsequent sample-path analysis. Finally, the decision maker is allowed to delete up to a time-average rate of p.

2.3. *Initial sample path.* Let $\{Q^0(t):t\in\mathbb{R}_+\}$ be the continuous-time queue length process, where $Q^0(t)\in\mathbb{Z}_+$ is the queue length at time t if *no deletion* is applied at any time. We say that an *event* occurs at time t if there is either an arrival, or a generation of service token, at time t. Let T_n, $n\in\mathbb{N}$, be the time of the nth event in the system. Denote by $\{Q^0[n]:n\in\mathbb{Z}_+\}$ the embedded discrete-time process of $\{Q^0(t)\}$, where $Q^0[n]$ is the length of the queue sampled immediately after the nth event,[6]

$$Q^0[n]=Q^0(T_n-),\qquad n\in\mathbb{N}$$

with the initial condition $Q^0[0]=0$. It is well known that Q^0 is a random walk on $\mathbb{Z}_+$, such that for all $x_1,x_2\in\mathbb{Z}_+$ and $n\in\mathbb{Z}_+$,

$$\mathbb{P}\big(Q^0[n+1]=x_2\,|\,Q^0[n]=x_1\big)=\begin{cases}\dfrac{\lambda}{\lambda+1-p}, & x_2-x_1=1,\\[2mm] \dfrac{1-p}{\lambda+1-p}, & x_2-x_1=-1,\\[2mm] 0, & \text{otherwise,}\end{cases}\tag{2.1}$$

if $x_1>0$ and

$$\mathbb{P}\big(Q^0[n+1]=x_2\,|\,Q^0[n]=x_1\big)=\begin{cases}\dfrac{\lambda}{\lambda+1-p}, & x_2-x_1=1,\\[2mm] \dfrac{1-p}{\lambda+1-p}, & x_2-x_1=0,\\[2mm] 0, & \text{otherwise,}\end{cases}\tag{2.2}$$

if $x_1=0$. Note that, when $\lambda>1-p$, the random walk Q^0 is transient.

The process Q^0 contains *all relevant information* in the arrival and service processes, and will be the main object of study of this paper. We will refer to Q^0 as the *initial sample path* throughout the paper, to distinguish it from sample paths obtained after deletions have been made.

[6]The notation $f(x-)$ denotes the right-limit of f at x: $f(x-)=\lim_{y\downarrow x}f(y)$. In this particular context, the values of $Q^0[n]$ are well defined, since the sample paths of Poisson processes are right-continuous-with-left-limits (RCLL) almost surely.

2.4. *Deletion policies*. Since a deletion can only take place when there is an arrival, it suffices to define the locations of deletions with respect to the discrete-time process $\{Q^0[n] : n \in \mathbb{Z}_+\}$, and throughout, our analysis will focus on discrete-time queue length processes unless otherwise specified. Let $\Phi(Q)$ be the locations of all arrivals in a discrete-time queue length process Q, that is,

$$\Phi(Q) = \{n \in \mathbb{N} : Q[n] > Q[n-1]\}$$

and for any $M \subset \mathbb{Z}_+$, define the counting process $\{I(M,n) : n \in \mathbb{N}\}$ associated with M as[7]

$$I(M,n) = |\{1,\dots,n\} \cap M|. \tag{2.3}$$

DEFINITION 1 (Feasible deletion sequence). The sequence $M = \{m_i\}$ is said to be a feasible deletion sequence with respect to a discrete-time queue length process, Q^0, if all of the following hold:

(1) All elements in M are unique, so that at most one deletion occurs at any slot.

(2) $M \subset \Phi(Q^0)$, so that a deletion occurs only when there is an arrival.

(3)

$$\limsup_{n\to\infty} \frac{1}{n} I(M,n) \le \frac{p}{\lambda + (1-p)} \qquad \text{a.s.} \tag{2.4}$$

so that the time-average deletion rate is at most p.

In general, M is also allowed to be a finite set.

The denominator $\lambda + (1-p)$ in equation (2.4) is due to the fact that the total rate of events in the system is $\lambda + (1-p)$.[8] Analogously, the deletion rate in continuous time is defined by

$$r_d = (\lambda + 1 - p) \cdot \limsup_{n\to\infty} \frac{1}{n} I(M,n). \tag{2.5}$$

The impact of a deletion sequence to the evolution of the queue length process is formalized in the following definition.

DEFINITION 2 (Deletion maps). Fix an initial queue length process $\{Q^0[n] : n \in \mathbb{N}\}$ and a corresponding feasible deletion sequence $M = \{m_i\}$.

(1) The *point-wise deletion map* $D_P(Q^0, m)$ outputs the resulting process after a deletion is made to Q^0 in slot m. Let $Q' = D_P(Q^0, m)$. Then

$$Q'[n] = \begin{cases} Q^0[n] - 1, & n \ge m \text{ and } Q^0[t] > 0 \ \forall t \in \{m,\dots,n\}; \\ Q^0[n], & \text{otherwise}, \end{cases} \tag{2.6}$$

[7] $|X|$ denotes the cardinality of X.

[8] This is equal to the total rate of jumps in $A(\cdot)$ and $S(\cdot)$.

(2) the *multi-point deletion map* $D(Q^0, M)$ outputs the resulting process after all deletions in the set M are made to Q^0. Define Q^i recursively as $Q^i = D_P(Q^{i-1}, m_i)$, $\forall i \in \mathbb{N}$. Then $Q^\infty = D(Q^0, M)$ is defined as the point-wise limit

$$Q^\infty[n] = \lim_{i \to \min\{|M|, \infty\}} Q^i[n] \qquad \forall n \in \mathbb{Z}_+. \tag{2.7}$$

The definition of the point-wise deletion map reflects the earlier assumption that the service time of a job only depends on the speed of the server at the moment and is independent of the job's identity; see Section 2. Note also that the value of $Q^\infty[n]$ depends only on the total number of deletions before n [equation (2.6)], which is at most n, and the limit in equation (2.7) is justified. Moreover, it is not difficult to see that the order in which the deletions are made has no impact on the resulting sample path, as stated in the lemma below. The proof is omitted.

LEMMA 1. *Fix an initial sample path Q^0, and let M and $\widetilde{M}$ be two feasible deletion sequences that contain the same elements. Then $D(Q^0, M) = D(Q^0, \widetilde{M})$.*

We next define the notion of a deletion policy that outputs a deletion sequence based on the (limited) knowledge of an initial sample path Q^0. Informally, a deletion policy is said to be w-lookahead if it makes its deletion decisions based on the knowledge of Q^0 up to w units of time into the future (in continuous time).

DEFINITION 3 (w-lookahead deletion policies). Fix $w \in \mathbb{R}_+ \cup \{\infty\}$. Let $\mathcal{F}_t = \sigma(Q^0(s); s \le t)$ be the natural filtration induced by $\{Q^0(t) : t \in \mathbb{R}_+\}$ and $\mathcal{F}_\infty = \bigcup_{t \in \mathbb{Z}_+} \mathcal{F}_t$. A w-predictive deletion policy is a mapping, $\pi : \mathbb{Z}_+^{\mathbb{R}_+} \to \mathbb{N}^\infty$, such that:

(1) $M = \pi(Q^0)$ is a feasible deletion sequence a.s.;
(2) $\{n \in M\}$ is $\mathcal{F}_{T_n + w}$ measurable, for all $n \in \mathbb{N}$.

We will denote by Π_w the family of all w-lookahead deletion policies.

The parameter w in Definition 3 captures the amount of information that the deletion policy has about the future:

(1) When $w = 0$, all deletion decisions are made solely based on the knowledge of the system up to the current time frame. We will refer to Π_0 as *online policies*.
(2) When $w = \infty$, the entire sample path of Q^0 is revealed to the decision maker at $t = 0$. We will refer to Π_∞ as *offline policies*.
(3) We will refer to Π_w, $0 < w < \infty$, as policies with a *lookahead window of size w*.

2.5. *Performance measure.* Given a discrete-time queue length process Q and $n \in \mathbb{N}$, denote by $S(Q, n) \in \mathbb{Z}_+$ the partial sum

$$S(Q, n) = \sum_{k=1}^{n} Q[k]. \tag{2.8}$$

DEFINITION 4 (Average post-deletion queue length). Let Q^0 be an initial queue length process. Define $C(p, \lambda, \pi) \in \mathbb{R}_+$ as the expected average queue length after applying a deletion policy π,

$$C(p, \lambda, \pi) = \mathbb{E}\left(\limsup_{n\to\infty} \frac{1}{n} S(Q_\pi^\infty, n)\right), \tag{2.9}$$

where $Q_\pi^\infty = D(Q^0, \pi(Q^0))$, and the expectation is taken over all realizations of Q^0 and the randomness used by π internally, if any.

REMARK (Delay versus queue length). By Little's law, the long-term average waiting time of a typical customer in the queue is equal to the long-term average queue length divided by the arrival rate (independent of the service discipline of the server). Therefore, if our goal is to minimize the average waiting time of the jobs that remain after deletions, it suffices to use $C(p, \lambda, \pi)$ as a performance metric in order to judge the effectiveness of a deletion policy π. In particular, denote by $T_{\text{all}} \in \mathbb{R}_+$ the time-average queueing delay experienced by all jobs, where deleted jobs are assumed to have a delay of zero, then $\mathbb{E}(T_{\text{all}}) = \frac{1}{\lambda} C(p, \lambda, \pi)$, and hence the average queue length and delay coincide in the heavy-traffic regime, as $\lambda \to 1$. With an identical argument, it is easy to see that the average delay among *admitted* jobs, T_{adt}, satisfies $\mathbb{E}(T_{\text{adt}}) = \frac{1}{\lambda - r_d} C(p, \lambda, \pi)$, where r_d is the continuous-time deletion rate under π. Therefore, we may use the terms "delay" and "average queue length" interchangeably in the rest of the paper, with the understanding that they represent essentially the same quantity up to a constant.

Finally, we define the notion of an optimal delay within a family of policies.

DEFINITION 5 (Optimal delay). Fix $w \in \mathbb{R}_+$. We call $C^*_{\Pi_w}(p, \lambda)$ the optimal delay in Π_w, where

$$C^*_{\Pi_w}(p, \lambda) = \inf_{\pi \in \Pi_w} C(p, \lambda, \pi). \tag{2.10}$$

3. Summary of main results. We state the main results of this paper in this section, whose proofs will be presented in Sections 5–7.

3.1. *Optimal delay for online policies.*

DEFINITION 6 (Threshold policies). We say that π^L_{th} is an L-threshold policy, if a job arriving at time t is deleted if and only if the queue length at time t is greater or equal to L.

The following theorem shows that the class of threshold policies achieves the optimal heavy-traffic delay scaling in Π_0.

THEOREM 1 (Optimal online policies). *Fix* $p \in (0,1)$, *and let*

$$L(p,\lambda) = \left\lceil \log_{\lambda/(1-p)} \frac{p}{1-\lambda} \right\rceil.$$

Then:

(1) $\pi_{\text{th}}^{L(p,\lambda)}$ *is feasible for all* $\lambda \in (1-p,1)$.

(2) $\pi_{\text{th}}^{L(p,\lambda)}$ *is asymptotically optimal in* Π_0 *as* $\lambda \to 1$,

$$C\big(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)}\big) \sim C^*_{\Pi_0}(p,\lambda) \sim \log_{1/(1-p)} \frac{1}{1-\lambda} \qquad \text{as } \lambda \to 1.$$

PROOF. See Section 5. □

3.2. *Optimal delay for offline policies.* Given the sample path of a random walk Q, let $U(Q,n)$ the number of slots till Q reaches the level $Q[n]-1$ after slot n:

$$U(Q,n) = \inf\big\{j \geq 1 : Q[n+j] = Q[n]-1\big\}. \tag{3.1}$$

DEFINITION 7 (No-job-left-behind policy[9]). Given an initial sample path Q^0, the no-job-left-behind policy, denoted by π_{NOB}, deletes all arrivals in the set Ψ, where

$$\Psi = \big\{n \in \Phi(Q^0) : U(Q^0,n) = \infty\big\}. \tag{3.2}$$

We will refer to the deletion sequence generated by π_{NOB} as $M^\Psi = \{m_i^\Psi : i \in \mathbb{N}\}$, where $M^\Psi = \Psi$.

In other words, π_{NOB} would delete a job arriving at time t if and only if the initial queue length process never returns to below the current level in the future, which also implies that

$$Q^0[n] \geq Q^0\big[m_i^\Psi\big] \qquad \forall i \in \mathbb{N}, n \geq m_i^\Psi. \tag{3.3}$$

Examples of the π_{NOB} policy being applied to a particular sample path are given in Figures 5 and 6 (illustration), as well as in Figure 7 (simulation).

It turns out that the delay performance of π_{NOB} is about as good as we can hope for in heavy traffic, as is formalized in the next theorem.

[9]The reason for choosing this name will be made in clear in Section 4.1, using the "stack" interpretation of this policy.

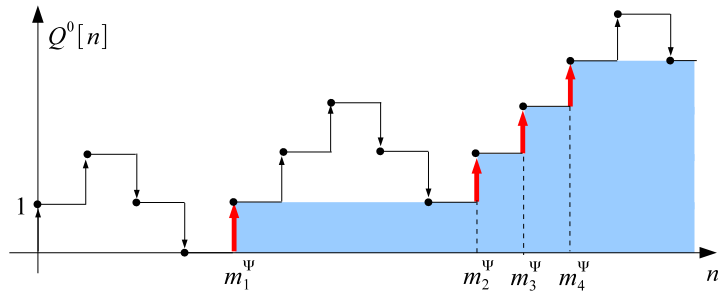

FIG. 5. *Illustration of applying* π_{NOB} *to an initial sample path,* Q^0*, where the deletions are marked by bold red arrows.*

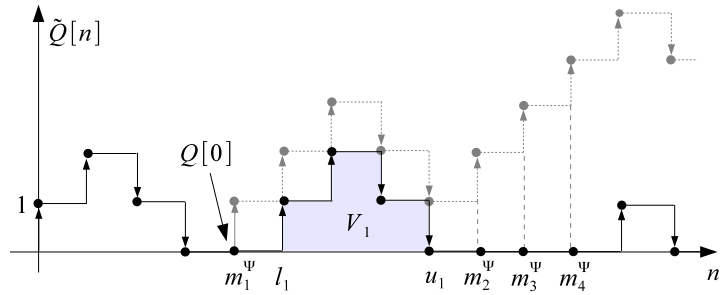

FIG. 6. *The solid lines depict the resulting sample path,* $\widetilde{Q} = D(Q^0, M^\Psi)$*, after applying* π_{NOB} *to* Q^0*.*

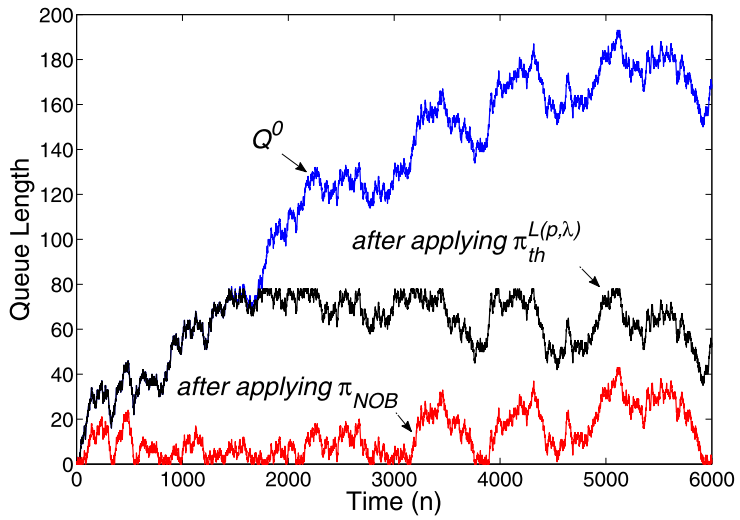

FIG. 7. *Example sample paths of* Q^0 *and those obtained after applying* $\pi_{\mathrm{th}}^{L(p,\lambda)}$ *and* π_{NOB} *to* Q^0*, with* $p = 0.05$ *and* $\lambda = 0.999$*.*

THEOREM 2 (Optimal offline policies). *Fix* $p \in (0, 1)$.

(1) π_{NOB} *is feasible for all* $\lambda \in (1-p, 1)$ *and*[10]

$$C(p, \lambda, \pi_{\mathrm{NOB}}) = \frac{1-p}{\lambda - (1-p)}. \tag{3.4}$$

(2) π_{NOB} *is asymptotically optimal in* Π_∞ *as* $\lambda \to 1$,

$$\lim_{\lambda \to 1} C(p, \lambda, \pi_{\mathrm{NOB}}) = \lim_{\lambda \to 1} C^*_{\Pi_\infty}(p, \lambda) = \frac{1-p}{p}.$$

PROOF. See Section 6. □

REMARK 1 (Heavy-traffic "delay collapse"). It is perhaps surprising to observe that the heavy-traffic scaling essentially "collapses" under π_{NOB}: the average queue length converges to a finite value, $\frac{1-p}{p}$, as $\lambda \to 1$, which is in sharp contrast with the optimal scaling of $\sim \log_{1/(1-p)} \frac{1}{1-\lambda}$ for the online policies, given by Theorem 1; see Figure 4 for an illustration of this difference. A "stack" interpretation of the no-job-left-behind policy (Section 4.1.1) will help us understand intuitively why such a drastic discrepancy exists between the online and offline heavy-traffic scaling behaviors.

Also, as a by-product of Theorem 2, observe that the heavy-traffic limit scales, in p, as

$$\lim_{\lambda \to 1} C^*_{\Pi_\infty}(p, \lambda) \sim \frac{1}{p} \qquad \text{as } p \to 0. \tag{3.5}$$

This is consistent with an intuitive notion of "flexibility": delay should degenerate as the system's ability to redirect away jobs diminishes.

REMARK 2 (Connections to branching processes and Erdős–Rényi random graphs). Let $d < 1 < c$ satisfy $de^{-d} = ce^{-c}$. Consider a Galton–Watson birth process in which each node has Z children, where Z is Poisson with mean c. Conditioning on the finiteness of the process gives a Galton–Watson process where Z is Poisson with mean d. This occurs in the classical analysis of the Erdős–Rényi random graph $G(n, p)$ with $p = c/n$. There will be a giant component and the deletion of that component gives a random graph $G(m, q)$ with $q = d/m$. As a rough analogy, π_{NOB} deletes those nodes that would be in the giant component.

[10]It is easy to see that π_{NOB} is not a very efficient deletion policy for relatively small values of λ. In fact, $C(p, \lambda, \pi_{\mathrm{NOB}})$ is a *decreasing* function of λ. This problem can be fixed by injecting into the arrival process an Poisson process of "dummy jobs" of rate $1 - \lambda - \varepsilon$, so that the total rate of arrival is $1 - \varepsilon$, where $\varepsilon \approx 0$. This reasoning implies that $(1-p)/p$ is a uniform upper-bound of $C^*_{\Pi_\infty}(p, \lambda)$ for all $\lambda \in (0, 1)$.

3.3. *Policies with a finite lookahead window.* In practice, infinite prediction into the future is certainly too much to ask for. In this section, we show that a natural modification of π_{NOB} allows for the *same delay* to be achieved, using only a *finite* lookahead window, whose length, $w(\lambda)$, increases to infinity as $\lambda \to 1$.[11]

Denote by $w \in \mathbb{R}_+$ the size of the lookahead window in continuous time, and $W(n) \in \mathbb{Z}_+$ the window size in the discrete-time embedded process Q^0, starting from slot n. Letting T_n be the time of the nth event in the system, then

$$W(n) = \sup\{k \in \mathbb{Z}_+ : T_{n+k} \le T_n + w\}. \tag{3.6}$$

For $x \in \mathbb{N}$, define the set of indices

$$U(Q, n, x) = \inf\bigl\{j \in \{1, \ldots, x\} : Q[n + j] = Q[n] - 1\bigr\}. \tag{3.7}$$

DEFINITION 8 (w-no-job-left-behind policy). Given an initial sample path Q^0 and $w > 0$, the w-no-job-left-behind policy, denoted by π^w_{NOB}, deletes all arrivals in the set Ψ^w, where

$$\Psi^w = \bigl\{n \in \Phi\bigl(Q^0\bigr) : U\bigl(Q^0, n, W(n)\bigr) = \infty\bigr\}.$$

It is easy to see that π^w_{NOB} is simply π_{NOB} applied within the confinement of a finite window: a job at t is deleted if and only if the initial queue length process does not return to below the current level *within the next w units of time*, assuming no further deletions are made. Since the window is finite, it is clear that $\Psi^w \supset \Psi$ for any $w < \infty$, and hence $C(p, \lambda, \pi^w_{\mathrm{NOB}}) \le C(p, \lambda, \pi_{\mathrm{NOB}})$ for all $\lambda \in (1 - p)$. The only issue now becomes that of feasibility: by making decision only based on a finite lookahead window, we may end up deleting at a rate greater than p.

The following theorem summarizes the above observations and gives an upper bound on the appropriate window size, w, as a function of λ.[12]

THEOREM 3 (Optimal delay scaling with finite lookahead). *Fix $p \in (0, 1)$. There exists $C > 0$, such that if*

$$w(\lambda) = C \cdot \log \frac{1}{1 - \lambda},$$

then $\pi^{w(\lambda)}_{\mathrm{NOB}}$ is feasible and

$$C\bigl(p, \lambda, \pi^{w(\lambda)}_{\mathrm{NOB}}\bigr) \le C(p, \lambda, \pi_{\mathrm{NOB}}) = \frac{1 - p}{\lambda - (1 - p)}. \tag{3.8}$$

[11] In a way, this is not entirely surprising, since the π_{NOB} leads to a deletion rate of $\lambda - (1 - p)$, and there is an additional $p - [\lambda - (1 - p)] = 1 - \lambda$ unused deletion rate that can be exploited.

[12] Note that Theorem 3 implies Theorem 2 and is hence stronger.

Since $C^*_{\Pi_{w(\lambda)}}(p,\lambda) \geq C^*_{\Pi_\infty}(p,\lambda)$ *and* $C^*_{\Pi_{w(\lambda)}}(p,\lambda) \leq C(p,\lambda,\pi^{w(\lambda)}_{\mathrm{NOB}})$, *we also have that*

$$\lim_{\lambda\to 1} C^*_{\Pi_{w(\lambda)}}(p,\lambda) = \lim_{\lambda\to 1} C^*_{\Pi_\infty}(p,\lambda) = \frac{1-p}{p}. \tag{3.9}$$

PROOF. See Section 7.1. □

3.3.1. *Delay-information duality.* Theorem 3 says that one can attain the same heavy-traffic delay performance as the optimal offline algorithm if the size of the lookahead window scales as $\mathcal{O}(\log \frac{1}{1-\lambda})$. Is this the minimum amount of future information necessary to achieve the same (or comparable) heavy-traffic delay limit as the optimal offline policy? We conjecture that this is the case, in the sense that there exists a matching lower bound, as follows.

CONJECTURE 1. *Fix* $p \in (0,1)$. *If* $w(\lambda) \ll \log \frac{1}{1-\lambda}$ *as* $\lambda \to 1$, *then*

$$\limsup_{\lambda\to 1} C^*_{\Pi_{w(\lambda)}}(p,\lambda) = \infty.$$

In other words, "delay collapse" can occur only if $w(\lambda) = \Theta(\log \frac{1}{1-\lambda})$.

If the conjecture is proven, it would imply a *sharp transition* in the system's heavy-traffic delay scaling behavior, around the critical "threshold" of $w(\lambda) = \Theta(\log \frac{1}{1-\lambda})$. It would also imply the existence of a symmetric dual relationship between *future information* and *queueing delay*: $\Theta(\log \frac{1}{1-\lambda})$ amount of information is required to achieve a finite delay limit, and one has to suffer $\Theta(\log \frac{1}{1-\lambda})$ in delay, if only finite amount of future information is available.

Figure 8 summarizes the main results of this paper from the angle of the delay-information duality. The dotted line segment marks the unknown regime and the sharp transition at its right endpoint reflects the view of Conjecture 1.

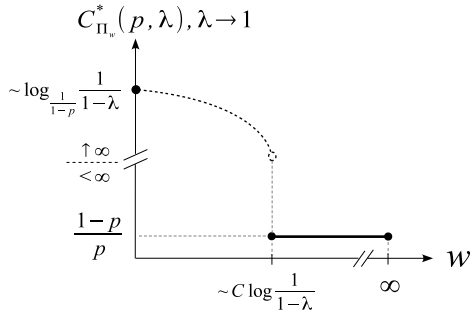

FIG. 8. *"Delay vs. Information." Best achievable heavy traffic delay scaling as a function of the size of the lookahead window,* w. *Results presented in this paper are illustrated in the solid lines and circles, and the gray dotted line depicts our conjecture of the unknown regime of* $0 < w(\lambda) \lesssim \log(\frac{1}{1-\lambda})$.

4. Interpretations of π_{NOB}. We present two equivalent ways of describing the no-job-left-behind policy π_{NOB}. The *stack interpretation* helps us derive asymptotic deletion rate of π_{NOB} in a simple manner, and illustrates the superiority of π_{NOB} compared to an online policy. Another description of π_{NOB} using time-reversal shows us that the set of deletions made by π_{NOB} can be calculated efficiently in linear time (with respect to the length of the time horizon).

4.1. *Stack interpretation.* Suppose that the service discipline adopted by the server is that of last-in-first-out (LIFO), where the it always fetches a task that has arrived the latest. In other words, the queue works as a *stack*. Suppose that we first simulate the stack without any deletion. It is easy to see that, when the arrival rate λ is greater than the service rate $1-p$, there will be a growing set of jobs at the bottom of the stack that will *never* be processed. Label all such jobs as "left-behind." For example, Figure 5 shows the evolution of the queue over time, where all "left-behind" jobs are colored with a blue shade. One can then verify that the policy π_{NOB} given in Definition 7 is equivalent to deleting all jobs that are labeled "left-behind," hence the namesake "No Job Left Behind." Figure 6 illustrates applying π_{NOB} to a sample path of Q^0, where the ith job to be deleted is precisely the ith job among all jobs that would have never been processed by the server under a LIFO policy.

One advantage of the stack interpretation is that it makes obvious the fact that the deletion rate induced by π_{NOB} is equal to $\lambda-(1-p)<p$, as illustrated in the following lemma.

LEMMA 2. *For all $\lambda>1-p$, the following statements hold*:

(1) *With probability one, there exists $T<\infty$, such that every service token generated after time T is matched with some job. In other words, the server never idles after some finite time.*

(2) *Let $Q=D(Q^0,M^\Psi)$. We have*

$$\limsup_{n\to\infty}\frac{1}{n}I\bigl(M^\Psi,n\bigr)\le\frac{\lambda-(1-p)}{\lambda+1-p}\qquad a.s., \tag{4.1}$$

which implies that π_{NOB} is feasible for all $p\in(0,1)$ and $\lambda\in(1-p,1)$.

PROOF. See Appendix A.1. □

4.1.1. *"Anticipation" vs. "reaction."* Some geometric intuition from the stack interpretation shows that the power of π_{NOB} essentially stems from being highly *anticipatory*. Looking at Figure 5, one sees that the jobs that are "left behind" at the bottom of the stack correspond to those who arrive during the intervals where the initial sample path Q^0 is taking a consecutive "upward hike." In other words, π_{NOB} begins to delete jobs when it anticipates that the arrivals are *just about to* get

intense. Similarly, a job in the stack will be "served" if Q^0 curves down eventually in the future, which corresponds π_{NOB}'s stopping deleting jobs as soon as it anticipates that the next few arrivals can be handled by the server alone. In sharp contrast is the nature of the optimal online policy, $\pi_{\mathrm{th}}^{L(p,\lambda)}$, which is by definition "reactionary" and begins to delete only when the current queue length has already reached a high level. The differences in the resulting sample paths are illustrated via simulations in Figure 7. For example, as Q^0 continues to increase during the first 1000 time slots, π_{NOB} begins deleting immediately after $t = 0$, while no deletion is made by $\pi_{\mathrm{th}}^{L(p,\lambda)}$ during this period.

As a rough analogy, the offline policy starts to delete *before* the arrivals get busy, but the online policy can only delete *after* the burst in arrival traffic has been realized, by which point it is already "too late" to fully contain the delay. This explains, to certain extend, why π_{NOB} is capable of achieving "delay collapse" in the heavy-traffic regime (i.e., a finite limit of delay as $\lambda \to 1$, Theorem 2), while the delay under even the best online policy diverges to infinity as $\lambda \to 1$ (Theorem 1).

4.2. *A linear-time algorithm for* π_{NOB}. While the offline deletion problem serves as a nice abstraction, it is impossible to actually store information about the *infinite* future in practice, even if such information is available. A natural finite-horizon version of the offline deletion problem can be posed as follows: given the values of Q^0 over the first N slots, where N finite, one would like to compute the set of deletions made by π_{NOB},

$$M_N^\Psi = M^\Psi \cap \{1, \ldots, N\}$$

assuming that $Q^0[n] > Q^0[N]$ for all $n \geq N$. Note that this problem also arises in computing the sites of deletions for the π_{NOB}^w policy, where one would replace N with the length of the lookahead window, w.

We have the following algorithm, which identifies all slots on which a new "minimum" (denoted by the variable S) is achieved in Q^0, when viewed in the *reverse* order of time.

A linear-time algorithm for π_{NOB}

$S \leftarrow Q^0[N]$ and $M_N^\Psi \leftarrow \varnothing$
for $n = N$ down to 1 **do**
  if $Q^0[n] < S$ **then**
    $M_N^\Psi \leftarrow M_N^\Psi \cup \{n+1\}$
    $S \leftarrow Q^0[n]$
  else
    $M_N^\Psi \leftarrow M_N^\Psi$
  end if
end for
return M_N^Ψ

It is easy to see that the running time of the above algorithm scales linearly with respect to the length of the time horizon, N. Note that this is not the unique linear-time algorithm. In fact, one can verify that the simulation procedure used in describing the stack interpretation of π_{NOB} (Section 4), which keeps track of which jobs would eventually be served, is itself a linear-time algorithm. However, the time-reverse version given here is arguably more intuitive and simpler to describe.

5. Optimal online policies. Starting from this section and through Section 7, we present the proofs of the results stated in Section 3.

We begin with showing Theorem 1, by formulating the online problem as a Markov decision problem (MDP) with an average cost constraint, which then enables us to use existing results to characterize the form of optimal policies. Once the family of threshold policies has been shown to achieve the optimal delay scaling in Π_0 under heavy traffic, the exact form of the scaling can be obtained in a fairly straightforward manner from the steady-state distribution of a truncated birth–death process.

5.1. *A Markov decision problem formulation.* Since both the arrival and service processes are Poisson, we can formulate the problem of finding an optimal policy in Π_0 as a continuous-time Markov decision problem with an average-cost constraint, as follows. Let $\{Q(t) : t \in \mathbb{R}_+\}$ be the resulting continuous-time queue length process after applying some policy in Π_0 to Q^0. Let T_k be the kth upward jump in Q and τ_k the length of the kth inter-jump interval, $\tau_k = T_k - T_{k-1}$. The task of a deletion policy, $\pi \in \Pi_0$, amounts to choosing, for each of the inter-jump intervals, a *deletion action*, $a_k \in [0, 1]$, where the value of a_k corresponds to the probability that the next arrival during the current inter-jump interval will be deleted. Define R and K to be the *reward* and *cost* functions of an inter-jump interval, respectively,

$$R(Q_k, a_k, \tau_k) = -Q_k \cdot \tau_k, \tag{5.1}$$

$$K(Q_k, a_k, \tau_k) = \lambda(1 - a_k)\tau_k, \tag{5.2}$$

where $Q_k = Q(T_k)$. The corresponding MDP seeks to maximize the time-average reward

$$\overline{R}_\pi = \liminf_{n\to\infty} \frac{\mathbb{E}_\pi(\sum_{k=1}^n R(Q_k, a_k, \tau_k))}{\mathbb{E}_\pi(\sum_{k=1}^n \tau_k)} \tag{5.3}$$

while obeying the average-cost constraint

$$\overline{C}_\pi = \limsup_{n\to\infty} \frac{\mathbb{E}_\pi(\sum_{k=1}^n K(Q_k, a_k, \tau_k))}{\mathbb{E}_\pi(\sum_{k=1}^n \tau_k)} \le p. \tag{5.4}$$

To see why this MDP solves our deletion problem, observe that $\overline{R}_\pi$ is the negative of the time-average queue length, and $\overline{C}_\pi$ is the time-average deletion rate.

It is well known that the type of constrained MDP described above admits an optimal policy that is stationary [1], which means that the action a_k depends solely on the current state, Q_k, and is independent of the time index k. Therefore, it suffices to describe π using a sequence, $\{b_q : q \in \mathbb{Z}_+\}$, such that $a_k = b_q$ whenever $Q_k = q$. Moreover, when the state space is finite,[13] stronger characterizations of the b_q's have been obtained for a family of reward and cost functions under certain regularity assumptions (Hypotheses 2.7, 3.1 and 4.1 in [5]), which ours do satisfy [equations (5.1) and (5.2)]. Theorem 1 will be proved using the next-known result (adapted from Theorem 4.4 in [5]):

LEMMA 3. *Fix p and λ, and let the buffer size B be finite. There exists an optimal stationary policy, $\{b_q^*\}$, of the form*

$$b_q^* = \begin{cases} 1, & q < L^* - 1, \\ \xi, & q = L^* - 1, \\ 0, & q \geq L^* \end{cases}$$

for some $L^* \in \mathbb{Z}_+$ and $\xi \in [0, 1]$.

5.2. *Proof of Theorem 1.* In words, Lemma 3 states that the optimal policy admits a "quasi-threshold" form: it deletes the next arrival when $Q(t) \geq L^*$, admits when $Q(t) < L^* - 1$, and admits with probability ξ when $Q(t) = L^* - 1$. Suppose, for the moment, that the statements of Lemma 3 also hold when the buffer size is infinite, an assumption to be justified by the end of the proof. Denoting by π_p^* the stationary optimal policy associated with $\{b_q^*\}$, when the constraint on the average of deletion is p [equation (5.4)]. The evolution of $Q(t)$ under π_p^* is that of a birth–death process truncated at state L^*, with the transition rates given in Figure 9, and the time-average queue length is equal to the expected queue length in steady state. Using standard calculations involving the steady-state distribution of the induced Markov process, it is not difficult to verify that

$$C\big(p, \lambda, \pi_{\text{th}}^{L^*-1}\big) \leq C\big(p, \lambda, \pi_p^*\big) \leq C\big(p, \lambda, \pi_{\text{th}}^{L^*}\big), \tag{5.5}$$

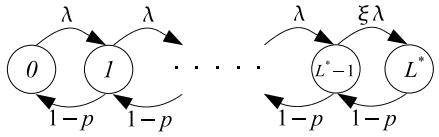

FIG. 9. *The truncated birth–death process induced by π_p^*.*

[13] This corresponds to a finite buffer size in our problem, where one can assume that the next arrival is automatically deleted when the buffer is full, independent of the value of a_k.

where L^* is defined as in Lemma 3, and $C(p,\lambda,\pi)$ is the time-average queue length under policy π, defined in equation (2.9).

Denote by $\{\mu_i^L : i \in \mathbb{N}\}$ the steady-state probability of the queue length being equal to i, under a threshold policy π_{th}^L. Assuming $\lambda \neq 1-p$, standard calculations using the balancing equations yield

$$\mu_i^L = \left(\frac{\lambda}{1-p}\right)^i \cdot \left(\frac{1-(\lambda/(1-p))}{1-(\lambda/(1-p))^{L+1}}\right) \qquad \forall 1 \le i \le L \tag{5.6}$$

and $\mu_i^L = 0$ for all $i \ge L+1$. The time-average queue length is given by

$$\begin{aligned} C(p,\lambda,\pi_{\text{th}}^L) &= \sum_{i=1}^{L} i \cdot \mu_i^L \\ &= \frac{\theta}{(\theta-1)(\theta^{L+1}-1)} \cdot \left[1-\theta^L + L\theta^L(\theta-1)\right], \end{aligned} \tag{5.7}$$

where $\theta = \frac{\lambda}{1-p}$. Note that when $\lambda > 1-p$, μ_i^L is decreasing with respect to L for all $i \in \{0,1,\ldots,L\}$ [equation (5.6)], which implies that the time-average queue length is monotonically increasing in L, that is,

$$\begin{aligned} &C(p,\lambda,\pi_{\text{th}}^{L+1}) - C(p,\lambda,\pi_{\text{th}}^L) \\ &\quad = (L+1)\cdot\mu_{L+1}^{L+1} + \sum_{i=0}^{L} i \cdot \left(\mu_i^{L+1} - \mu_i^L\right) \\ &\quad \ge (L+1)\cdot\mu_{L+1}^{L+1} + L \cdot \left(\sum_{i=0}^{L} \mu_i^{L+1} - \mu_i^L\right) \\ &\quad = (L+1)\cdot\mu_{L+1}^{L+1} + L \cdot \left(1 - \mu_i^{L+1} - 1\right) \\ &\quad = \mu_{L+1}^{L+1} > 0. \end{aligned} \tag{5.8}$$

It is also easy to see that, fixing p, since we have that $\theta > 1+\delta$ for all λ sufficiently close to 1, where $\delta > 0$ is a fixed constant, we have

$$C(p,\lambda,\pi_{\text{th}}^L) = \left(\frac{\theta^{L+1}}{\theta^{L+1}-1}\right)L - \frac{\theta}{\theta-1}\cdot\frac{\theta^L-1}{\theta^{L+1}-1} \sim L \qquad \text{as } L \to \infty. \tag{5.9}$$

Since deletions only occur when $Q(t)$ is in state L, from equation (5.6), the average rate of deletions in continuous time under π_{th}^L is given by

$$r_d(p,\lambda,\pi_{\text{th}}^L,) = \lambda \cdot \pi_L = \lambda \cdot \left(\frac{\lambda}{1-p}\right)^L \cdot \left(\frac{1-(\lambda/(1-p))}{1-(\lambda/(1-p))^{L+1}}\right). \tag{5.10}$$

Define

$$L(x,\lambda) = \min\left\{L \in \mathbb{Z}_+ : r_d(p,\lambda,\pi_{\text{th}}^L,) \le x\right\}, \tag{5.11}$$

that is, $L(x,\lambda)$ is the smallest L for which π_{th}^{L} remains feasible, given a deletion rate constraint of x. Using equations (5.10) and (5.11) to solve for $L(p,\lambda)$, we obtain, after some algebra,

$$L(p,\lambda) = \left\lceil \log_{\lambda/(1-p)} \frac{p}{1-\lambda} \right\rceil \sim \log_{1/(1-\tilde{p})} \frac{1}{1-\lambda} \qquad \text{as } \lambda \to 1 \tag{5.12}$$

and, by combining equation (5.12) and equation (5.9) with $L = L(p,\lambda)$, we have

$$C\big(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)}\big) \sim L(p,\lambda) \sim \log_{1/(1-p)} \frac{1}{1-\lambda} \qquad \text{as } \lambda \to 1. \tag{5.13}$$

By equations (5.8) and (5.11), we know that $\pi_{\text{th}}^{L(p,\lambda)}$ achieves the minimum average queue length among all feasible threshold policies. By equation (5.5), we must have that

$$C\big(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)-1}\big) \le C\big(p,\lambda,\pi_p^*\big) \le C\big(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)}\big). \tag{5.14}$$

Since Lemma 3 only applies when $B < \infty$, equation (5.14) holds whenever the buffer size, B, is greater than $L(p,\lambda)$, but finite. We next extend equation (5.14) to the case of $B = \infty$. Denote by ν_p^* a stationary optimal policy, when $B = \infty$ and the constraint on average deletion rate is equal to p [equation (5.4)]. The upper bound on $C(p,\lambda,\pi_p^*)$ in equation (5.14) automatically holds for $C(p,\lambda,\nu_p^*)$, since $C(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)})$ is still feasible when $B = \infty$. It remains to show a lower bound of the form

$$C\big(p,\lambda,\nu_p^*\big) \ge C\big(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)-2}\big), \tag{5.15}$$

when $B = \infty$, which, together with the upper bound, will have implied that the scaling of $C(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)})$ [equation (5.13)] carries over to ν_p^*,

$$C\big(p,\lambda,\nu_p^*\big) \sim C\big(p,\lambda,\pi_{\text{th}}^{L(p,\lambda)}\big) \sim \log_{1/(1-p)} \frac{1}{1-\lambda} \qquad \text{as } \lambda \to 1, \tag{5.16}$$

thus proving Theorem 1.

To show equation (5.15), we will use a straightforward truncation argument that relates the performance of an optimal policy under $B = \infty$ to the case of $B < \infty$. Denote by $\{b_q^*\}$ the deletion probabilities of a stationary optimal policy, ν_p^*, and by $\{b_q^*(B')\}$ the deletion probabilities for a truncated version, $\nu_p^*(B')$, with

$$b_q^*(B') = \mathbb{I}(q \le B') \cdot b_q^*$$

for all $q \ge 0$. Since ν_p^* is optimal and yields the minimum average queue length, it is without loss of generality to assume that the Markov process for $Q(t)$ induced by ν_p^* is positive recurrent. Denoting by $\{\mu_i^*\}$ and $\{\mu_i^*(B')\}$ the steady-state probability of queue length being equal to i under ν_p^* and $\nu_p^*(B')$, respectively, it follows from the positive recurrence of $Q(t)$ under ν_p and some algebra, that

$$\lim_{B' \to \infty} \mu_i^*(B') = \mu_i^* \tag{5.17}$$

for all $i \in \mathbb{Z}_+$ and

$$\lim_{B' \to \infty} C(p, \lambda, \nu_p^*(B')) = C(p, \lambda, \nu_p^*). \tag{5.18}$$

By equation (5.17) and the fact that $b_i^*(B') = b_i^*$ for all $0 \le i \le B'$, we have that[14]

$$\begin{aligned} \lim_{B' \to \infty} r_d(p, \lambda, \nu_p^*(B')) &= \lim_{B' \to \infty} \lambda \sum_{i=0}^{\infty} \mu_i^*(B') \cdot (1 - b_i^*(B')) \\ &= r_d(p, \lambda, \nu_p^*) \\ &\le p. \end{aligned} \tag{5.19}$$

It is not difficult to verify, from the definition of $L(p, \lambda)$ [equation (5.11)], that

$$\lim_{\delta \to 0} L(p + \delta, \lambda) \ge L(p, \lambda) - 1,$$

for all p, λ. For all $\delta > 0$, choose B' to be sufficiently large, so that

$$C(p, \lambda, \nu_p^*(B')) \le C(p, \lambda, \nu_p^*) + \delta, \tag{5.20}$$

$$L(\lambda, r_d(p, \lambda, \nu_p^*(B'))) \ge L(p, \lambda) - 1. \tag{5.21}$$

Let $p' = r_d(p, \lambda, \nu_p^*(B'))$. Since $b_i^*(B') = 0$ for all $i \ge B' + 1$, by equation (5.21) we have

$$C(p, \lambda, \nu_p^*(B')) \ge C(p, \lambda, \pi_{p'}^*), \tag{5.22}$$

where π_p^* is the optimal stationary policy given in Lemma 3 under any the finite buffer size $B > B'$. We have

$$\begin{aligned} & C(p, \lambda, \nu_p^*) + \delta \\ & \overset{(a)}{\ge} C(p, \lambda, \nu_p^*(B')) \\ & \overset{(b)}{\ge} C(p, \lambda, \pi_{p'}^*) \\ & \overset{(c)}{\ge} C(p, \lambda, \pi_{\text{th}}^{L(p', \lambda) - 1}) \\ & \overset{(d)}{\ge} C(p, \lambda, \pi_{\text{th}}^{L(p, \lambda) - 2}), \end{aligned} \tag{5.23}$$

where the inequalities (a) through (d) follow from equations (5.20), (5.22), (5.14) and (5.21), respectively. Since equation (5.23) holds for all $\delta > 0$, we have proven equation (5.15). This completes the proof of Theorem 1.

[14] Note that, in general, $r_d(p, \lambda, \nu_p^*(B'))$ could be greater than p, for any finite B'.

6. Optimal offline policies. We prove Theorem 2 in this section, which is completed in two parts. In the first part (Section 6.2), we give a full characterization of the sample path resulted by applying π_{NOB} (Proposition 1), which turns out to be a *recurrent* random walk. This allows us to obtain the steady-state distribution of the queue length under π_{NOB} in closed-form. From this, the expected queue length, which is equal to the time-average queue length, $C(p, \lambda, \pi_{\mathrm{NOB}})$, can be easily derived and is shown to be $\frac{1-p}{\lambda-(1-p)}$. Several side results we obtain along this path will also be used in subsequent sections.

The second part of the proof (Section 6.3) focuses on showing the heavy-traffic optimality of π_{NOB} among the class of all feasible offline policies, namely, that $\lim_{\lambda\to 1} C(p, \lambda, \pi_{\mathrm{NOB}}) = \lim_{\lambda\to 1} C^*_{\Pi_\infty}(p, \lambda)$, which, together with the first part, proves Theorem 2 (Section 6.4). The optimality result is proved using a sample-path-based analysis, by relating the resulting queue length sample path of π_{NOB} to that of a greedy deletion rule, which has an optimal deletion performance over a *finite* time horizon, $\{1, \ldots, N\}$, given any initial sample path. We then show that the discrepancy between π_{NOB} and the greedy policy, in terms of the resulting time-average queue length after deletion, diminishes almost surely as $N \to \infty$ and $\lambda \to 1$ (with the two limits taken in this order). This establishes the heavy-traffic optimality of π_{NOB}.

6.1. *Additional notation.* Define $\widetilde{Q}$ as the resulting queue length process after applying π_{NOB}

$$\widetilde{Q} = D(Q^0, M^\Psi)$$

and Q as the shifted version of $\widetilde{Q}$, so that Q starts from the first deletion in $\widetilde{Q}$,

$$Q[n] = \widetilde{Q}\big[n + m_1^\Psi\big], \qquad n \in \mathbb{Z}_+. \tag{6.1}$$

We say that $B = \{l, \ldots, u\} \subset \mathbb{N}$ is a *busy period* of Q if

$$Q[l-1] = Q[u] = 0 \quad \text{and} \quad Q[n] > 0 \qquad \text{for all } n \in \{l, \ldots, u-1\}. \tag{6.2}$$

We may write $B_j = \{l_j, \ldots, u_j\}$ to mean the jth busy period of Q. An example of a busy period is illustrated in Figure 6.

Finally, we will refer to the set of slots between two adjacent deletions in Q (note the offset of m_1),

$$E_i = \{m_i^\Psi - m_1^\Psi, m_i^\Psi + 1 - m_1^\Psi, \ldots, m_{i+1}^\Psi - 1 - m_1^\Psi\} \tag{6.3}$$

as the ith *deletion epoch*.

6.2. *Performance of the no-job-left-behind policy.* For simplicity of notation, throughout this section, we will denote by $M = \{m_i : i \in \mathbb{N}\}$ the deletion sequence generated by applying π_{NOB} to Q^0, when there is no ambiguity (as opposed to using M^Ψ and m_i^Ψ). The following lemma summarizes some important properties of Q which will be used repeatedly.

LEMMA 4. *Suppose* $1 > \lambda > 1 - p > 0$. *The following hold with probability one*:

(1) *For all* $n \in \mathbb{N}$, *we have* $Q[n] = Q^0[n + m_1] - I(M, n + m_1)$.
(2) *For all* $i \in \mathbb{N}$, *we have* $n = m_i - m_1$, *if and only if*

$$Q[n] = Q[n-1] = 0 \tag{6.4}$$

with the convention that $Q[-1] = 0$. *In other words, the appearance of two consecutive zeros in* Q *is equivalent to having a deletion on the second zero.*
(3) $Q[n] \in \mathbb{Z}_+$ *for all* $n \in \mathbb{Z}_+$.

PROOF. See Appendix A.2 □

The next proposition is the main result of this subsection. It specifies the probability law that governs the evolution of Q.

PROPOSITION 1. $\{Q[n] : n \in \mathbb{Z}_+\}$ *is a random walk on* $\mathbb{Z}_+$, *with* $Q[0] = 0$, *and, for all* $n \in \mathbb{N}$ *and* $x_1, x_2 \in \mathbb{Z}_+$,

$$\mathbb{P}\big(Q[n+1] = x_2 | Q[n] = x_2\big) = \begin{cases} \dfrac{1-p}{\lambda + 1 - p}, & x_2 - x_1 = 1, \\ \dfrac{\lambda}{\lambda + 1 - p}, & x_2 - x_1 = -1, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 > 0$ *and*

$$\mathbb{P}\big(Q[n+1] = x_2 | Q[n] = x_1\big) = \begin{cases} \dfrac{1-p}{\lambda + 1 - p}, & x_2 - x_1 = 1, \\ \dfrac{\lambda}{\lambda + 1 - p}, & x_2 - x_1 = 0, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 = 0$.

PROOF. For a sequence $\{X[n] : n \in \mathbb{N}\}$ and $s, t \in \mathbb{N}$, $s \le t$, we will use the shorthand

$$X_s^t = \{X[s], \ldots, X[t]\}.$$

Fix $n \in N$, and a sequence $(q_1, \ldots, q_n) \subset \mathbb{Z}_+^n$. We have

$$\begin{aligned} &\mathbb{P}\big(Q[n] = q[n] | Q_1^{n-1} = q_1^{n-1}\big) \\ &= \sum_{k=1}^{n} \sum_{\substack{t_1, \ldots, t_k, \\ t_k \le n-1+t_1}} \mathbb{P}\big(Q[n] = q[n] | Q_1^{n-1} = q_1^{n-1}, m_1^k = t_1^k, m_{k+1} \ge n + t_1\big) \\ &\qquad \times \mathbb{P}\big(m_1^k = t_1^k, m_{k+1} \ge n + t_1 | Q_1^{n-1} = q_1^{n-1}\big). \end{aligned} \tag{6.5}$$

Restricting to the values of t_i's and $q[i]$'s under which the summand is nonzero, the first factor in the summand can be written as

$$\begin{aligned}
&\mathbb{P}(Q[n]=q[n]|Q_1^{n-1}=q_1^{n-1}, m_1^k=t_1^k, m_{k+1}\geq n+t_1)\\
&\quad=\mathbb{P}(\widetilde{Q}[n+m_1]=q[n]|\widetilde{Q}_{m_1+1}^{m_1+n-1}=q_1^{n-1}, m_1^k=t_1^k, m_{k+1}\geq n+t_1)\\
&\quad\overset{(a)}{=}\mathbb{P}\Big(Q^0[n+t_1]=q[n]+k|Q^0[s+t_1]=q[s]+I\big(\{t_i\}_{i=1}^k, s+t_1\big),\\
&\qquad\qquad\forall 1\leq s\leq n-1 \text{ and } \min_{r\geq n+t_1} Q^0[r]\geq k\Big)\\
&\quad\overset{(b)}{=}\mathbb{P}\Big(Q^0[n+t_1]=q[n]+k|Q^0[n-1+t_1]=q[n-1]+k\\
&\qquad\qquad\text{and } \min_{r\geq n+t_1} Q^0[r]\geq k\Big),
\end{aligned}\tag{6.6}$$

where $\widetilde{Q}$ was defined in equation (6.1). Step (a) follows from Lemma 4 and the fact that $t_k\leq n-1+t_1$, and (b) from the Markov property of Q^0 and the fact that the events $\{\min_{r\geq n+t_1} Q^0[r]\geq k\}$, $\{Q^0[n+t_1]=q[n]+k\}$ and their intersection, depend only on the values of $\{Q^0[s]:s\geq n+t_1\}$, and are hence independent of $\{Q^0[s]:1\leq s\leq n-2+t_1\}$ conditional on the value of $Q^0[t_1+n-1]$.

Since the process Q lives in $\mathbb{Z}_+$ (Lemma 4), it suffices to consider the case of $q[n]=q[n-1]+1$, and show that

$$\begin{aligned}
&\mathbb{P}\Big(Q^0[n+t_1]=q[n-1]+1+k|Q^0[n-1+t_1]=q[n-1]+k\\
&\qquad\qquad\text{and } \min_{r\geq n+t_1} Q^0[r]\geq k\Big)\\
&\quad=\frac{1-p}{\lambda+1-p}
\end{aligned}\tag{6.7}$$

for all $q[n-1]\in\mathbb{Z}_+$. Since $Q[m_i-m_1]=Q[m_i-1-m_1]=0$ for all i (Lemma 4), the fact that $q[n]=q[n-1]+1>0$ implies that

$$n<m_{k+1}-1+m_1.\tag{6.8}$$

Moreover, since $Q^0[m_{k+1}-1]=k$ and $n<m_{k+1}-1+m_1$, we have that

$$q[n]>0 \qquad \text{implies } Q^0[t]=k \qquad \text{for some } t\geq n+1+m_1.\tag{6.9}$$

We consider two cases, depending on the value of $q[n-1]$.

Case 1: $q[n-1]>0$. Using the same argument that led to equation (6.9), we have that

$$q[n-1]>0 \qquad \text{implies } Q^0[t]=k \qquad \text{for some } t\geq n+m_1.\tag{6.10}$$

It is important to note that, despite the similarity of their conclusions, equations (6.9) and (6.10) are different in their assumptions (i.e., $q[n]$ versus $q[n-1]$). We have

$$\begin{aligned}
&\mathbb{P}\Big(Q^0[n+t_1] = q[n-1]+1+k \,|\, Q^0[n-1+t_1] = q[n-1]+k \\
&\qquad\qquad \text{and } \min_{r\geq n+t_1} Q^0[r] \geq k\Big) \\
&\overset{\text{(a)}}{=} \mathbb{P}\Big(Q^0[n+t_1] = q[n-1]+1+k \,|\, Q^0[n-1+t_1] = q[n-1]+k \\
&\qquad\qquad \text{and } \min_{r\geq n+t_1} Q^0[r] = k\Big) \\
&\overset{\text{(b)}}{=} \mathbb{P}\Big(Q^0[2] = q[n-1]+1 \,|\, Q^0[1] = q[n-1] \text{ and } \min_{r\geq 2} Q^0[r] = 0\Big) \\
&\overset{\text{(c)}}{=} \frac{1-p}{\lambda+1-p},
\end{aligned} \tag{6.11}$$

where (a) follows from equation (6.10), (b) from the stationary and space-homogeneity of the Markov chain Q^0 and (c) from the following well-known property of a transient random walk conditional to returning to zero:

LEMMA 5. *Let $\{X[n]: n \in \mathbb{N}\}$ be a random walk on $\mathbb{Z}_+$, such that for all $x_1, x_2 \in \mathbb{Z}_+$ and $n \in \mathbb{N}$,*

$$\mathbb{P}\big(X[n+1] = x_2 \,|\, X[n] = x_2\big) = \begin{cases} q, & x_2 - x_1 = 1, \\ 1-q, & x_2 - x_1 = -1, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 > 0$ and

$$\mathbb{P}\big(X[n+1] = x_2 \,|\, X[n] = x_1\big) = \begin{cases} q, & x_2 - x_1 = 1, \\ 1-q, & x_2 - x_1 = 0, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 = 0$, where $q \in (\frac{1}{2}, 1)$. Then for all $x_1, x_2 \in \mathbb{Z}_+$ and $n \in \mathbb{N}$,

$$\mathbb{P}\Big(X[n+1] = x_2 \,|\, X[n] = x_1, \min_{r\geq n+1} X[r] = 0\Big) = \begin{cases} 1-q, & x_2 - x_1 = 1, \\ q, & x_2 - x_1 = -1, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 > 0$ and

$$\mathbb{P}\Big(X[n+1] = x_2 \,|\, X[n] = x_1, \min_{r\geq n+1} X[r] = 0\Big) = \begin{cases} 1-q, & x_2 - x_1 = 1, \\ q, & x_2 - x_1 = 0, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 = 0$. In other words, conditional on the eventual return to 0 and before it happens, a transient random walk obeys the same probability law as a random walk with the reversed one-step transition probability.

PROOF. See Appendix A.3. □

Case 2: $q[n-1] = 0$ We have

$$\begin{aligned}
&\mathbb{P}\Big(Q^0[n+t_1] = q[n-1] + 1 + k \,|\, Q^0[n-1+t_1] = q[n-1] + k \\
&\qquad\qquad \text{and } \min_{r \geq n+t_1} Q^0[r] \geq k\Big) \\
&\overset{(a)}{=} \mathbb{P}\Big(Q^0[n+t_1] = 1 + k \text{ and } \min_{r > n+t_1} Q^0[r] = k \,|\, Q^0[n-1+t_1] = k \\
&\qquad\qquad \text{and } \min_{r \geq n+t_1} Q^0[r] \geq k\Big) \\
&\overset{(b)}{=} \mathbb{P}\Big(Q^0[2] = 2 \text{ and } \min_{r>2} Q^0[r] = 1 \,|\, Q^0[1] = 1 \text{ and } \min_{r \geq 2} Q^0[r] \geq 1\Big), \\
&\triangleq x,
\end{aligned} \tag{6.12}$$

where (a) follows from equation (6.9) [note its difference with equation (6.10)], and (b) from the stationarity and space-homogeneity of Q^0, and the assumption that $k \geq 1$ [equation (6.5)].

Since equations (6.11) and (6.12) hold for all $x_1, k \in \mathbb{Z}_+$ and $n \geq m_1 + 1$, by equation (6.5), we have that

$$\begin{aligned}
&\mathbb{P}(Q[n] = q[n] \,|\, Q_1^{n-1} = q_1^{n-1}) \\
&= \begin{cases} \dfrac{1-p}{\lambda + 1 - p}, & q[n] - q[n-1] = 1, \\ \dfrac{\lambda}{\lambda + 1 - p}, & q[n] - q[n-1] = -1, \\ 0, & \text{otherwise}, \end{cases}
\end{aligned} \tag{6.13}$$

if $q[n-1] > 0$ and

$$\mathbb{P}(Q[n] = q[n] \,|\, Q_1^{n-1} = q_1^{n-1}) = \begin{cases} x, & q[n] - q[n-1] = 1, \\ 1 - x, & q[n] - q[n-1] = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{6.14}$$

if $q[n-1] = 0$, where x represents the value of the probability in equation (6.12). Clearly, $Q[0] = Q^0[m_1] = 0$. We next show that x is indeed equal to $\frac{1-p}{\lambda+1-p}$, which will have proven Proposition 1.

One can in principle obtain the value of x by directly computing the probability in line (b) of equation (6.12), which can be quite difficult to do. Instead, we will use an indirect approach that turns out to be computationally much simpler: we will relate x to the rate of deletion of π_{NOB} using renewal theory, and then solve for x. As a by-product of this approach, we will also get a better understanding of an important regenerative structure of π_{NOB} [equation (6.20)], which will be useful for the analysis in subsequent sections.

By equations (6.13) and (6.14), Q is a positive recurrent Markov chain, and $Q[n]$ converges to a well-defined steady-state distribution, $Q[\infty]$, as $n \to \infty$. Letting $\pi_i = \mathbb{P}(Q[\infty] = i)$, it is easy to verify via the balancing equations that

$$\pi_i = \pi_0 \frac{x(\lambda + 1 - p)}{\lambda} \cdot \left(\frac{1-p}{\lambda}\right)^{i-1} \qquad \forall i \geq 1 \tag{6.15}$$

and since $\sum_{i \geq 0} \pi_i = 1$, we obtain

$$\pi_0 = \frac{1}{1 + x \cdot (\lambda + 1 - p)/(\lambda - (1-p))}. \tag{6.16}$$

Since the chain Q is also irreducible, the limiting fraction of time that Q spends in state 0 is therefore equal to π_0,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}(Q[t] = 0) = \pi_0 = \frac{1}{1 + x \cdot (\lambda + 1 - p)/(\lambda - (1-p))}. \tag{6.17}$$

Next, we would like to know many of these visits to state 0 correspond to a deletion. Recall the notion of a busy period and deletion epoch, defined in equations (6.2) and (6.3), respectively. By Lemma 4, n corresponds to a deletion if any only if $Q[n] = Q[n-1] = 0$. Consider a deletion in slot m_i. If $Q[m_i + 1] = 0$, then $m_i + 1$ also corresponds to a deletion, that is, $m_i + 1 = m_{i+1}$. If instead $Q[m_i + 1] = 1$, which happens with probability x, the fact that $Q[m_{i+1} - 1] = 0$ implies that there exists at least one busy period, $\{l, \ldots, u\}$, between m_i and m_{i+1}, with $l = m_i$ and $u \leq m_{i+1} - 1$. At the end of this period, a new busy period starts with probability x and so on. In summary, a deletion epoch E_i consists of the slot $m_i - m_1$, plus N_i busy periods, where the N_i are i.i.d., with[15]

$$N_1 \stackrel{d}{=} \mathrm{Geo}(1-x) - 1 \tag{6.18}$$

and hence

$$|E_i| = 1 + \sum_{j=1}^{N_i} B_{i,j}, \tag{6.19}$$

where $\{B_{i,j} : i, j \in \mathbb{N}\}$ are i.i.d. random variables, and $B_{i,j}$ corresponds to the length of the jth busy period in the ith epoch.

[15] $\mathrm{Geo}(p)$ denotes a geometric random variable with mean $\frac{1}{p}$.

Define $W[t] = (Q[t], Q[t+1])$, $t \in \mathbb{Z}_+$. Since Q is Markov, $W[t]$ is also a Markov chain, taking values in $\mathbb{Z}_+^2$. Since a deletion occurs in slot t if and only if $Q[t] = Q[t-1] = 0$ (Lemma 4), $|E_i|$ corresponds to excursion times between two adjacent visits of W to the state $(0,0)$, and hence are i.i.d. Using the elementary renewal theorem, we have

$$\lim_{n\to\infty} \frac{1}{n} I(M,n) = \frac{1}{\mathbb{E}(|E_1|)} \qquad \text{a.s.} \tag{6.20}$$

and by viewing each visit of W to $(0,0)$ as a renewal event and using the fact that exactly one deletion occurs within a deletion epoch. Denoting by R_i the number of visits to the state 0 within E_i, we have that $R_i = 1 + N_i$. Treating R_i as the reward associated with the renewal interval E_i, we have, by the time-average of a renewal reward process (cf. Theorem 6, Chapter 3, [10]), that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}\big(Q[t]=0\big) = \frac{\mathbb{E}(R_1)}{\mathbb{E}(|E_1|)} = \frac{\mathbb{E}(N_1)+1}{\mathbb{E}(|E_1|)} \qquad \text{a.s.} \tag{6.21}$$

by treating each visit of Q to $(0,0)$ as a renewal event. From equations (6.20) and (6.21), we have

$$\frac{\lim_{n\to\infty}(1/n) I(M,n)}{\lim_{n\to\infty}(1/n)\sum_{t=1}^{n} \mathbb{I}(Q[t]=0)} = \frac{1}{\mathbb{E}(N_1)} = 1-x. \tag{6.22}$$

Combining equations (4.1), (6.17) and (6.22), and the fact that $\mathbb{E}(N_1) = \mathbb{E}(\text{Geo}(1-x)) - 1 = \frac{1}{1-x} - 1$, we have

$$\frac{\lambda - (1-p)}{\lambda + 1 - p} \cdot \left[1 + x \cdot \frac{\lambda + 1 - p}{\lambda - (1-p)}\right] = 1 - x, \tag{6.23}$$

which yields

$$x = \frac{1-p}{\lambda + 1 - p}. \tag{6.24}$$

This completes the proof of Proposition 1. □

We summarize some of the key consequences of Proposition 1 below, most of which are easy to derive using renewal theory and well-known properties of positive-recurrent random walks.

PROPOSITION 2. *Suppose that* $1 > \lambda > 1 - p > 0$, *and denote by* $Q[\infty]$ *the steady-state distribution of* Q.

(1) *For all* $i \in \mathbb{Z}_+$,

$$\mathbb{P}\big(Q[\infty] = i\big) = \left(1 - \frac{1-p}{\lambda}\right) \cdot \left(\frac{1-p}{\lambda}\right)^i. \tag{6.25}$$

(2) *Almost surely, we have that*

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} Q[i] = \mathbb{E}\big(Q[\infty]\big) = \frac{1-p}{\lambda-(1-p)}. \tag{6.26}$$

(3) *Let* $E_i = \{m_i^{\Psi}, m_i^{\Psi}+1, \ldots, m_{i+1}^{\Psi}-1, m_{i+1}^{\Psi}\}$. *Then the* $|E_i|$ *are i.i.d., with*

$$\mathbb{E}\big(|E_1|\big) = \frac{1}{\lim_{n\to\infty}(1/n)I(M^{\Psi},n)} = \frac{\lambda+1-p}{\lambda-(1-p)} \tag{6.27}$$

and there exists $a, b > 0$ *such that for all* $x \in \mathbb{R}_+$

$$\mathbb{P}\big(|E_1| \geq x\big) \leq a\cdot\exp(-b\cdot x). \tag{6.28}$$

(4) *Almost surely, we have that*

$$m_i^{\Psi} \sim \frac{1}{\mathbb{E}(|E_1|)}\cdot i = \frac{\lambda-(1-p)}{\lambda+1-p}\cdot i \tag{6.29}$$

as $i\to\infty$.

PROOF. Claim 1 follows from the well-known steady-state distribution of a random walk, or equivalently, the fact that $Q[\infty]$ has the same distribution as the steady-state number of jobs in an $M/M/1$ queue with traffic intensity $\rho = \frac{1-p}{\lambda}$. For Claim 2, since Q is an irreducible Markov chain that is positive recurrent, it follows that its time-average coincides with $\mathbb{E}(Q[\infty])$ almost surely.

The fact that E_i's are i.i.d. was shown in the discussion preceding equation (6.20) in the proof of Proposition 1. The value of $\mathbb{E}(|E_1|)$ follows by combining equations (4.1) and (6.20).

Let $B_{i,j}$ be the length of the jth busy period [defined in equation (6.2)] in E_i. By definition, $B_{1,1}$ is distributed as the time till the random walk Q reaches state 0, starting from state 1. We have

$$\mathbb{P}(B_{1,1}\geq x) \leq \mathbb{P}\left(\sum_{j=1}^{\lfloor x\rfloor} X_j \leq -1\right),$$

where the X_j's are i.i.d., with $\mathbb{P}(X_1=1) = \frac{1-p}{\lambda+1-p}$ and $\mathbb{P}(X_1=-1) = \frac{\lambda}{\lambda+1-p}$, which, by the Chernoff bound, implies an exponential tail bound for $\mathbb{P}(B_{1,1}\geq x)$, and in particular,

$$\lim_{\theta\downarrow 0} G_{B_{1,1}}(\theta) = 1. \tag{6.30}$$

By equation (6.19), the moment generating function for $|E_1|$ is given by

$$\begin{aligned} G_{|E_1|}(\varepsilon) &= \mathbb{E}\big(\exp\big(\varepsilon \cdot |E_1|\big)\big) \\ &= \mathbb{E}\left(\exp\left(\varepsilon \cdot \left(1 + \sum_{j=1}^{N_1} B_{1,j}\right)\right)\right) \\ &\overset{\text{(a)}}{=} \mathbb{E}\big(e^{\varepsilon}\big) \cdot \mathbb{E}\big(\exp\big(N_1 \cdot G_{B_{1,1}}(\varepsilon)\big)\big) \\ &= \mathbb{E}\big(e^{\varepsilon}\big) \cdot G_{N_1}\big(\ln\big(G_{B_{1,1}}(\varepsilon)\big)\big), \end{aligned} \tag{6.31}$$

where (a) follows from the fact that $\{N_1\} \cup \{B_{1,j} : j \in \mathbb{N}\}$ are mutually independent, and $G_{N_1}(x) = \mathbb{E}(\exp(x \cdot N_1))$. Since $N_1 \overset{d}{=} \text{Geo}(1-x) - 1$, $\lim_{x \downarrow 0} G_{N_1}(x) = 1$, and by equation (6.30), we have that $\lim_{\varepsilon \downarrow 0} G_{|E_1|}(\varepsilon) = 1$, which implies equation (6.28).

Finally, equation (6.29) follows from the third claim and the elementary renewal theorem. □

6.3. *Optimality of the no-job-left-behind policy in heavy traffic.* This section is devoted to proving the optimality of π_{NOB} as $\lambda \to 1$, stated in the second claim of Theorem 2, which we isolate here in the form of the following proposition.

PROPOSITION 3. *Fix $p \in (0,1)$. We have that*

$$\lim_{\lambda \to 1} C(p, \lambda, \pi_{\text{NOB}}) = \lim_{\lambda \to 1} C^*_{\Pi_\infty}(p, \lambda).$$

The proof is given at the end of this section, and we do so by showing the following:

(1) Over a finite horizon N and given a fixed number of deletions to be made, a greedy deletion rule is optimal in minimizing the post-deletion area under Q over $\{1, \ldots, N\}$.

(2) Any point of deletion chosen by π_{NOB} will also be chosen by the greedy policy, as $N \to \infty$.

(3) The fraction of points chosen by the greedy policy but not by π_{NOB} diminishes as $\lambda \to 1$, and hence the delay produced by π_{NOB} is the best possible, as $\lambda \to 1$.

Fix $N \in \mathbb{N}$. Let $S(Q, N)$ be the partial sum $S(Q, N) = \sum_{n=1}^{N} Q[n]$. For any sample path Q, denote by $\Delta(Q, n)$ the marginal decrease of area under Q over the horizon $\{1, \ldots, N\}$ by applying a deletion at slot n, that is,

$$\Delta_P(Q, N, n) = S(Q, N) - S\big(D_P(Q, n), N\big)$$

and, analogously,

$$\Delta\big(Q, N, M'\big) = S(Q, N) - S\big(D(Q, M'), N\big),$$

where M' is a deletion sequence.

We next define the notion of a greedy deletion rule, which constructs a deletion sequence by recursively adding the slot that leads to the maximum marginal decrease in $S(Q, N)$.

DEFINITION 9 (Greedy deletion rule). Fix an initial sample path Q^0 and $K, N \in \mathbb{N}$. The *greedy deletion rule* is a mapping, $G(Q^0, N, K)$, which outputs a finite deletion sequence $M^G = \{m_i^G : 1 \le i \le K\}$, given by

$$m_1^G \in \arg \max_{m \in \Phi(Q^0, N)} \Delta_P\big(Q^0, N, m\big),$$

$$m_k^G \in \arg \max_{m \in \Phi(Q^{k-1}, N)} \Delta_P\big(Q^{k-1}_{M^G}, N, m\big), \qquad 2 \le k \le K,$$

where $\Phi(Q, N) = \Phi(Q) \cap \{1, \ldots, N\}$ is the set of all locations in Q in the first N slots that can be deleted, and $Q^k_{M^G} = D(Q^0, \{m_i^G : 1 \le i \le k\})$. Note that we will allow $m_k^G = \infty$, if there is no more entry to delete [i.e., $\Phi(Q^{k-1}) \cap \{1, \ldots, N\} = \varnothing$].

We now state a key lemma that will be used in proving Theorem 2. It shows that over a finite horizon and for a finite number of deletions, the greedy deletion rule yields the maximum reduction in the area under the sample path.

LEMMA 6 (Dominance of greedy policy). *Fix an initial sample path Q^0, horizon $N \in \mathbb{N}$ and number of deletions $K \in \mathbb{N}$. Let M' be any deletion sequence with $I(M', N) = K$. Then*

$$S\big(D(Q^0, M'), N\big) \ge S\big(D\big(Q^0, M^G\big), N\big),$$

where $M^G = G(Q^0, N, K)$ is the deletion sequence generated by the greedy policy.

PROOF. By Lemma 1, it suffices to show that, for any sample path $\{Q[n] \in \mathbb{Z}_+ : n \in \mathbb{N}\}$ with $|Q[n+1] - Q[n]| = 1$ if $Q[n] > 0$ and $|Q[n+1] - Q[n]| \in \{0, 1\}$ if $Q[n] = 0$, we have

$$\begin{aligned} &S\big(D(Q, M'), N\big) \\ &\quad \ge \Delta_P\big(Q, N, m_1^G\big) + \min_{\substack{|\widetilde{M}| = k-1, \\ \widetilde{M} \subset \Phi(D(Q, m_1^G), N)}} S\big(D\big(Q^1_{M^G}, \widetilde{M}\big), N\big). \end{aligned} \tag{6.32}$$

By induction, this would imply that we should use the greedy rule at every step of deletion up to K. The following lemma states a simple monotonicity property. The proof is elementary, and is omitted.

LEMMA 7 (Monotonicity in deletions). *Let Q and Q' be two sample paths such that*

$$Q[n] \le Q'[n] \qquad \forall n \in \{1, \dots, N\}.$$

Then, for any $K \ge 1$,

$$\min_{\substack{|M|=K,\\ M\subset\Phi(Q,N)}} S\big(D(Q,M),N\big) \le \min_{\substack{|M|=K,\\ M\subset\Phi(Q',N)}} S\big(D(Q',M),N\big) \tag{6.33}$$

and, for any finite deletion sequence $M' \subset \Phi(Q,N)$,

$$\Delta\big(Q,N,M'\big) \ge \Delta\big(Q',N,M'\big). \tag{6.34}$$

Recall the definition of a busy period in equation (6.2). Let $J(Q,N)$ be the total number of busy periods in $\{Q[n]: 1 \le n \le N\}$, with the additional convention $Q[N+1] \stackrel{\triangle}{=} 0$ so that the last busy period always ends on N. Let $B_j = \{l_j, \dots, u_j\}$ be the jth busy period. It can be verified that a deletion in location n leads to a decrease in the value of $S(Q,N)$ that is no more than the width of the busy period to which n belongs; cf. Figure 6. Therefore, by definition, a greedy policy always seeks to delete in each step the first arriving job during a longest busy period in the current sample path, and hence

$$\Delta\big(Q,N,G(Q,N,1)\big) = \max_{1\le j\le J(Q,N)} |B_j|. \tag{6.35}$$

Let

$$\mathcal{J}^*(Q,N) = \arg \max_{1\le j\le J(Q,N)} |B_j|.$$

We consider the following cases, depending on whether M' chooses to delete any job in the busy periods in $\mathcal{J}^*(Q,N)$.

Case 1: $M' \cap (\bigcup_{j\in\mathcal{J}^*(Q,N)} B_j) \ne \varnothing$. If $l_{j^*} \in M'$ for some $j^* \in \mathcal{J}^*$, by equation (6.35), we can set m_1^G to l_{j^*}. Since $m_1^G \in M'$ and the order of deletions does not impact the final resulting delay (Lemma 1), we have that equation (6.32) holds, and we are done. Otherwise, choose $m^* \in M' \cap B_{j^*}$ for some $j^* \in \mathcal{J}^*$, and we have $m^* > l_{j^*}$. Let

$$Q' = D_P(Q, m^*) \quad \text{and} \quad \widehat{Q} = D_P(Q, l_{j^*}).$$

Since $Q[n] > 0$, $\forall n \in \{l_{j^*}, \dots, u_{j^*} - 1\}$, we have $\widehat{Q}[n] = Q[n] - 1 \le Q'[n]$, $\forall n \in \{l_{j^*}, \dots, u_{j^*} - 1\}$ and $Q'[n] = Q[n] = \widehat{Q}[n]$, $\forall n \notin \{l_{j^*}, \dots, u_{j^*} - 1\}$, which implies that

$$\widehat{Q}[n] \le Q'[n] \qquad \forall n \in \{1, \dots, N\}. \tag{6.36}$$

Equation (6.32) holds by combining equation (6.36) and equation (6.33) in Lemma 7, with $K = k - 1$.

Case 2: $M' \cap (\bigcup_{j \in \mathcal{J}^*(Q,N)} B_j) = \varnothing$. Let m^* be any element in M' and $Q' = D_P(Q, m^*)$. Clearly, $Q[n] \geq Q'[n]$ for all $n \in \{1, \ldots, N\}$, and by equation (6.34) in Lemma 7, we have that[16]

$$\Delta(Q, N, M' \setminus \{m^*\}) \geq \Delta(D_P(Q, m^*), N, M' \setminus \{m^*\}). \tag{6.37}$$

Since $M' \cap (\bigcup_{j \in \mathcal{J}^*(Q,N)} B_j) = \varnothing$, we have that

$$\Delta_P(D(Q, M' \setminus \{m^*\}), N, m_1^G) = \max_{1 \leq j \leq J(Q,N)} |B_j| > \Delta_P(Q, N, m^*). \tag{6.38}$$

Let $\widehat{M} = m_1^G \cup (M' \setminus \{m^*\})$, and we have that

$$\begin{aligned}
&S(D(Q, \widehat{M}), N) \\
&\quad = S(Q, N) - \Delta(Q, N, M' \setminus \{m^*\}) - \Delta_P(D(Q, M' \setminus \{m^*\}), N, m_1^G) \\
&\quad \overset{(a)}{\leq} S(Q, N) - \Delta(D_P(Q, m^*), N, M' \setminus \{m^*\}) \\
&\qquad - \Delta_P(D(Q, M' \setminus \{m^*\}), N, m_1^G) \\
&\quad \overset{(b)}{<} S(Q, N) - \Delta(D_P(Q, m^*), N, M' \setminus \{m^*\}) - \Delta_P(Q, N, m^*) \\
&\quad = S(D(Q, M'), N),
\end{aligned}$$

where (a) and (b) follow from equations (6.37) and (6.38), respectively, which shows that equation (6.32) holds (and in this case the inequality there is strict).

Cases 1 and 2 together complete the proof of Lemma 6. □

We are now ready to prove Proposition 3.

PROOF OF PROPOSITION 3. Lemma 6 shows that, for any fixed number of deletions over a finite horizon N, the greedy deletion policy (Definition 9) yields the smallest area under the resulting sample path, Q, over $\{1, \ldots, N\}$. The main idea of proof is to show that the area under Q after applying π_{NOB} is asymptotically the same as that of the greedy policy, as $N \to \infty$ and $\lambda \to 1$ (in this particular order of limits). In some sense, this means that the jobs in M^{Ψ} account for almost all of the delays in the system, as $\lambda \to 1$. The following technical lemma is useful.

LEMMA 8. *For a finite set $S \subset \mathbb{R}$ and $k \in \mathbb{N}$, define*

$$f(S, k) = \frac{\text{sum of the } k \text{ largest elements in } S}{|S|}.$$

Let $\{X_i : 1 \leq i \leq n\}$ be i.i.d. random variables taking values in $\mathbb{Z}_+$, where $\mathbb{E}(X_1) < \infty$. Then for any sequence of random variables $\{H_n : n \in \mathbb{N}\}$, with

[16] For finite sets A and B, $A \setminus B = \{a \in A : a \notin B\}$.

$H_n \lesssim \alpha n$ a.s. as $n \to \infty$ for some $\alpha \in (0,1)$, we have

$$\limsup_{n\to\infty} f\big(\{X_i : 1 \le i \le n\}, H_n\big) \le \mathbb{E}\big(X_1 \cdot \mathbb{I}\big(X_1 \ge \overline{F}^{-1}_{X_1}(\alpha)\big)\big) \qquad a.s., \tag{6.39}$$

where $\overline{F}^{-1}_{X_1}(y) = \min\{x \in \mathbb{N} : \mathbb{P}(X_1 \ge x) < y\}$.

PROOF. See Appendix A.4. □

Fix an initial sample path Q^0. We will denote by $M^\Psi = \{m_i^\Psi : i \in \mathbb{N}\}$ the deletion sequence generated by π_{NOB} on Q^0. Define

$$l(n) = n - \max_{1 \le i \le I(M^\Psi, n)} |E_i|, \tag{6.40}$$

where E_i is the ith deletion epoch of M^Ψ, defined in equation (6.3). Since $Q^0[n] \ge Q^0[m_i]$ for all $i \in \mathbb{N}$, it is easy to check that

$$\Delta_P\big(D\big(Q^0, \{m_j^\Psi : 1 \le j \le i-1\}\big), n, m_i^\Psi\big) = n - m_i^\Psi + 1$$

for all $i \in \mathbb{N}$. The function l was defined so that the first $I(M^\Psi, l(n))$ deletions made by a greedy rule over the horizon $\{1, \dots, n\}$ are exactly $\{1, \dots, l(n)\} \cap M^\Psi$. More formally, we have the following lemma.

LEMMA 9. *Fix $n \in \mathbb{N}$, and let $M^G = G(Q^0, n, I(M^\Psi, l(n)))$. Then $m_i^G = m_i^\Psi$, for all $i \in \{1, \dots, I(M^\Psi, l(n))\}$.*

Fix $K \in \mathbb{N}$, and an arbitrary feasible deletion sequence, $\widetilde{M}$, generated by a policy in Π_∞. We can write

$$\begin{aligned} I\big(\widetilde{M}, m_K^\Psi\big) &= I\big(M^\Psi, l\big(m_K^\Psi\big)\big) + \big(I\big(M^\Psi, m_K^\Psi\big) - I\big(M^\Psi, l\big(m_K^\Psi\big)\big)\big) \\ &\quad + \big(I\big(\widetilde{M}, m_K^\Psi\big) - I\big(M^\Psi, m_K^\Psi\big)\big) \\ &= I\big(M^\Psi, l\big(m_K^\Psi\big)\big) + \big(K - I\big(M^\Psi, l\big(m_K^\Psi\big)\big)\big) \\ &\quad + \big(I\big(\widetilde{M}, m_K^\Psi\big) - I\big(M^\Psi, m_K^\Psi\big)\big) \\ &= I\big(M^\Psi, l\big(m_K^\Psi\big)\big) + h(K), \end{aligned} \tag{6.41}$$

where

$$h(K) = \big(K - I\big(M^\Psi, l\big(m_K^\Psi\big)\big)\big) + \big(I\big(\widetilde{M}, m_K^\Psi\big) - I\big(M^\Psi, m_K^\Psi\big)\big). \tag{6.42}$$

We have the following characterization of h.

LEMMA 10. $h(K) \lesssim \frac{1-\lambda}{\lambda-(1-p)} \cdot K$, *as* $K \to \infty$, *a.s.*

PROOF. See Appendix A.5. □

Let

$$M^{G,n} = G\big(Q^0, n, I(\widetilde{M}, n)\big), \tag{6.43}$$

where the greedy deletion map G was defined in Definition 9. By Lemma 9 and the definition of $M^{G,n}$, we have that

$$M^{\Psi} \cap \{1, \dots, l(m_K^{\Psi})\} \subset M^{G,m_K^{\Psi}}. \tag{6.44}$$

Therefore, we can write

$$M^{G,m_K^{\Psi}} = \big(M^{\Psi} \cap \{1, \dots, l(m_K^{\Psi})\}\big) \cup \overline{M}_K^G, \tag{6.45}$$

where $\overline{M}_K^G \triangleq M^{G,m_K^{\Psi}} \setminus (M^{\Psi} \cap \{1, \dots, l(m_K^{\Psi})\})$. Since $|M^{G,m_K^{\Psi}}| = I(\widetilde{M}, m_K^{\Psi})$ by definition, by equation (6.41),

$$\big|\overline{M}_K^G\big| = h(K). \tag{6.46}$$

We have

$$\begin{aligned} &S\big(D(Q^0, M^{\Psi}), m_K^{\Psi}\big) - S\big(D(Q^0, \widetilde{M}), m_K^{\Psi}\big) \\ &\overset{(a)}{\leq} S\big(D(Q^0, M^{\Psi}), m_K^{\Psi}\big) - S\big(D(Q^0, M^{G,m_K^{\Psi}}), m_K^{\Psi}\big) \\ &\overset{(b)}{=} \Delta\big(D(Q^0, M^{\Psi}), m_K^{\Psi}, \overline{M}_K^G\big), \end{aligned} \tag{6.47}$$

where (a) is based on the dominance of the greedy policy over any finite horizon (Lemma 6), and (b) follows from equation (6.45).

Finally, we claim that there exists $g(x): \mathbb{R} \to \mathbb{R}_+$, with $g(x) \to 0$ as $x \to 1$, such that

$$\limsup_{K\to\infty} \frac{\Delta(D(Q^0, M^{\Psi}), m_K^{\Psi}, \overline{M}_K^G)}{m_K^{\Psi}} \leq g(\lambda) \qquad \text{a.s.} \tag{6.48}$$

Equations (6.47) and (6.48) combined imply that

$$\begin{aligned} C(p, \lambda, \pi_{\mathrm{NOB}}) &= \limsup_{K\to\infty} \frac{S(D(Q^0, M^{\Psi}), m_K^{\Psi})}{m_K^{\Psi}} \\ &\leq g(\lambda) + \limsup_{K\to\infty} \frac{S(D(Q^0, \widetilde{M}), m_K^{\Psi})}{m_K^{\Psi}}, \\ &= g(\lambda) + \limsup_{n\to\infty} \frac{S(D(Q^0, \widetilde{M}), n)}{n} \qquad \text{a.s.}, \end{aligned} \tag{6.49}$$

which shows that

$$C(p, \lambda, \pi_{\mathrm{NOB}}) \leq g(\lambda) + \inf_{\pi \in \Pi_\infty} C(p, \lambda, \pi).$$

Since $g(\lambda) \to 0$ as $\lambda \to 1$, this proves Proposition 3.

To show equation (6.48), denote by Q the sample path after applying π_{NOB},

$$Q = D\big(Q^0, M^{\Psi}\big)$$

and by V_i the area under Q within E_i,

$$V_i = \sum_{n=m_i^\Psi}^{m_{i+1}^\Psi - 1} Q[n].$$

An example of V_i is illustrated as the area of the shaded region in Figure 6. By Proposition 1, Q is a Markov chain, and so is the process $W[n] = (Q[n], Q[n+1])$. By Lemma 4, E_i corresponds to the indices between two adjacent returns of the chain W to state $(0,0)$. Since the ith return of a Markov chain to a particular state is a stopping time, it can be shown, using the strong Markov property of W, that the segments of Q, $\{Q[n] : n \in E_i\}$, are mutually independent and identically distributed among different values of i. Therefore, the V_i's are i.i.d. Furthermore,

$$\mathbb{E}(V_1) \overset{(a)}{\leq} \mathbb{E}\big(|E_1|^2\big) \overset{(b)}{<} \infty, \tag{6.50}$$

where (a) follows from the fact that $|Q[n+1] - Q[n]| \leq 1$ for all n, and hence $V_i \leq |E_i|^2$ for any sample path of Q^0, and (b) from the exponential tail bound on $\mathbb{P}(|E_1| \geq x)$, given in equation (6.28).

Since the value of Q on the two ends of E_i, m_i^Ψ and $m_{i+1}^\Psi - 1$, are both zero, each additional deletion within E_i cannot produce a marginal decrease of area under Q of more than V_i; cf. Figure 6. Therefore, the value of $\Delta(D(Q^0, M^\Psi), m_K^\Psi, \overline{M}_K^G)$ can be no greater than the sum of the $h(K)$ largest V_i's over the horizon $n \in \{1, \ldots, m_K^\Psi\}$. We have

$$\begin{aligned}
&\limsup_{K\to\infty} \frac{\Delta(D(Q^0, M^\Psi), m_K^\Psi, \overline{M}_K^G)}{m_K^\Psi} \\
&\quad = \limsup_{K\to\infty} f\big(\{V_i : 1 \leq i \leq K\}, h(K)\big) \cdot \frac{K}{m_K^\Psi} \\
&\quad \overset{(a)}{=} \limsup_{K\to\infty} f\big(\{V_i : 1 \leq i \leq K\}, h(K)\big) \cdot \frac{\lambda + 1 - p}{\lambda - (1-q)} \\
&\quad \overset{(b)}{=} \mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda - (1-p)}\right)\right)\right) \cdot \frac{\lambda + 1 - p}{\lambda - (1-q)},
\end{aligned} \tag{6.51}$$

where (a) follows from equation (6.29), and (b) from Lemmas 8 and 10. Since $\mathbb{E}(V_1) < \infty$, and $\overline{F}_{V_1}^{-1}(x) \to \infty$ as $x \to 0$, it follows that

$$\mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda - (1-p)}\right)\right)\right) \to 0$$

as $\lambda \to 1$. Equation (6.48) is proved by setting

$$g(\lambda) = \mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda - (1-p)}\right)\right)\right) \cdot \frac{\lambda + 1 - p}{\lambda - (1-q)}.$$

This completes the proof of Proposition 3. □

6.3.1. *Why not use greedy?* The proof of Proposition 3 relies on a sample-path-wise coupling to the performance of a greedy deletion rule. It is then only natural to ask: since the time horizon is indeed finite in all practical applications, why do not we simply use the greedy rule as the preferred offline policy, as opposed to π_{NOB}?

There are at least two reasons for focusing on π_{NOB} instead of the greedy rule. First, the structure of the greedy rule is highly global, in the sense that each deletion decision uses information of the entire sample path over the horizon. As a result, the greedy rule tells us little on how to design a good policy with a *fixed* lookahead window (e.g., Theorem 3). In contrast, the performance analysis of π_{NOB} in Section 6.2 reveals a highly *regenerative* structure: the deletions made by π_{NOB} essentially depend only on the dynamics of Q^0 in the same deletion epoch (the E_i's), and what happens beyond the current epoch becomes irrelevant. This is the key intuition that led to our construction of the finite-lookahead policy in Theorem 3. A second (and perhaps minor) reason is that of computational complexity. By a small sacrifice in performance, π_{NOB} can be efficiently implemented using a linear-time algorithm (Section 4.2), while it is easy to see that a naive implementation of the greedy rule would require super-linear complexity with respect to the length of the horizon.

6.4. *Proof of Theorem 2.* The fact that π_{NOB} is feasible follows from equation (4.1) in Lemma 2, that is,

$$\limsup_{n\to\infty} \frac{1}{n} I\bigl(M^{\Psi}, n\bigr) \le \frac{\lambda - (1-p)}{\lambda + 1 - p} < \frac{p}{\lambda + 1 - p} \qquad \text{a.s.}$$

Let $\{\widetilde{Q}[n] : n \in \mathbb{Z}_+\}$ be the resulting sample path after applying π_{NOB} to the initial sample path $\{Q^0[n] : n \in \mathbb{Z}_+\}$, and let

$$Q[n] = \widetilde{Q}\bigl[n + m_1^{\Psi}\bigr] \qquad \forall n \in \mathbb{N},$$

where m_1^{Ψ} is the index of the first deletion made by π_{NOB}. Since $\lambda > 1 - p$, the random walk Q^0 is transient, and hence $m_1^{\Psi} < \infty$ almost surely. We have that, almost surely,

$$\begin{aligned} C(p, \lambda, \pi_{\mathrm{NOB}}) &= \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \widetilde{Q}[i] \\ &= \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{m_1^{\Psi}} \widetilde{Q}[i] + \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Q[i] \\ &= \frac{1-p}{\lambda - (1-p)}, \end{aligned} \tag{6.52}$$

where the last equality follows from equation (6.26) in Proposition 2, and the fact that $m_1 < \infty$ almost surely. Letting $\lambda \to 1$ in equation (6.52) yields the finite limit of delay under heavy traffic,

$$\lim_{\lambda\to 1} C(p,\lambda,\pi_{\mathrm{NOB}}) = \lim_{\lambda\to 1} \frac{1-p}{\lambda-(1-p)} = \frac{1-p}{p}.$$

Finally, the delay optimality of π_{NOB} in heavy traffic was proved in Proposition 3, that is, that

$$\lim_{\lambda\to 1} C(p,\lambda,\pi_{\mathrm{NOB}}) = \lim_{\lambda\to 1} C^*_{\Pi_\infty}(p,\lambda).$$

This completes the proof of Theorem 2.

7. Policies with a finite lookahead.

7.1. *Proof of Theorem 3.* As pointed out in the discussion preceding Theorem 3, for any initial sample path and $w < \infty$, an arrival that is deleted under the π_{NOB} policy will also be deleted under π^w_{NOB}. Therefore, the delay guarantee for π_{NOB} (Theorem 2) carries over to $\pi^{w(\lambda)}_{\mathrm{NOB}}$, and for the rest of the proof, we will be focusing on showing that $\pi^{w(\lambda)}_{\mathrm{NOB}}$ is feasible under an appropriate scaling of $w(\lambda)$. We begin by stating an exponential tail bound on the distribution of the discrete-time predictive window, $W(\lambda,n)$, defined in equation (3.6),

$$W(\lambda,n) = \max\bigl\{k \in \mathbb{Z}_+ : T_{n+k} \le T_n + w(\lambda)\bigr\}.$$

It is easy to see that $\{W(\lambda,m_i^\Psi) : i \in \mathbb{N}\}$ are i.i.d., with $W(\lambda,m_1^\Psi)$ distributed as a Poisson random variable with mean $(\lambda+1-p)w(\lambda)$. Since

$$\mathbb{P}\bigl(W(\lambda,m_1^\Psi) \ge x\bigr) \le \mathbb{P}\left(\sum_{k=1}^{\lfloor w(\lambda)\rfloor} X_k\right),$$

where the X_k are i.i.d. Poisson random variables with mean $\lambda+(1-p)$, applying the Chernoff bound, we have that, there exist $c,d>0$ such that

$$\mathbb{P}\left(W(\lambda,m_1^\Psi) \ge \frac{\lambda+1-p}{2}\cdot w(\lambda)\right) \le c\cdot\exp\bigl(-d\cdot w(\lambda)\bigr) \tag{7.1}$$

for all $w(\lambda) > 0$.

We now analyze the deletion rate resulted by the $\pi^{w(\lambda)}_{\mathrm{NOB}}$ policy. For the pure purpose of analysis (as opposed to practical efficiency), we will consider a new deletion policy, denoted by $\sigma^{w(\lambda)}$, which can be viewed as a relaxation of $\pi^{w(\lambda)}_{\mathrm{NOB}}$.

DEFINITION 10. Fix $w \in \mathbb{R}_+$. The deletion policy σ^w is defined such that for each deletion epoch E_i, $i \in \mathbb{N}$:

(1) if $|E_i| \le W(\lambda, m_i^{\Psi})$, then only the first arrival of this epoch, namely, the arrival in slot m_i^{Ψ}, is deleted;
(2) otherwise, all arrivals within this epoch are deleted.

It is easy to verify that σ^w can be implemented with w units of look-ahead, and the set of deletions made by $\sigma^{w(\lambda)}$ is a strict superset of $\pi_{\mathrm{NOB}}^{w(\lambda)}$ almost surely. Hence, the feasibility of $\sigma^{w(\lambda)}$ will imply that of $\pi_{\mathrm{NOB}}^{w(\lambda)}$.

Denote by D_i the number of deletions made by $\sigma^{w(\lambda)}$ in the epoch E_i. By the construction of the policy, the D_i are i.i.d., and depend only on the length of E_i and the number of arrivals within. We have[17]

$$\begin{aligned}
\mathbb{E}(D_1) &\le 1 + \mathbb{E}\big[|E_i| \cdot \mathbb{I}\big(|E_i| \ge W\big(\lambda, m_i^{\Psi}\big)\big)\big] \\
&\le 1 + \mathbb{E}\left[|E_i| \cdot \mathbb{I}\left(|E_i| \ge \frac{\lambda + 1 - p}{2} \cdot w(\lambda)\right)\right] \\
&\quad + \mathbb{E}\big(|E_i|\big) \cdot \mathbb{P}\left(W\big(\lambda, m_i^{\Psi}\big) \le \frac{\lambda + 1 - p}{2} \cdot w(\lambda)\right) \\
&\le 1 + \left(\sum_{k=((\lambda+1-p)/2)\cdot w(\lambda)}^{\infty} k \cdot a \cdot \exp(-b \cdot k)\right) \\
&\quad + \frac{\lambda}{\lambda - (1-p)} \cdot c \cdot \exp\big(-d \cdot w(\lambda)\big) \\
&\overset{\text{(a)}}{\le} 1 + h \cdot w(\lambda) \cdot \exp\big(-l \cdot w(\lambda)\big)
\end{aligned} \tag{7.2}$$

for some $h, l > 0$, where (a) follows from the fact that $\sum_{k=n}^{\infty} k \cdot \exp(-b \cdot k) = \mathcal{O}(n \cdot \exp(-b \cdot n))$ as $n \to \infty$.

Since the D_i are i.i.d., using basic renewal theory, it is not difficult to show that the average rate of deletion in discrete time under the policy $\sigma^{w(\lambda)}$ is equal to $\frac{\mathbb{E}(D_1)}{\mathbb{E}(E_1)}$. In order for the policy to be feasible, one must have that

$$\frac{\mathbb{E}(D_1)}{\mathbb{E}(E_1)} = \frac{\mathbb{E}(D_1)}{\lambda} \le \frac{p}{\lambda + 1 - p}. \tag{7.3}$$

By equations (7.2) and (7.3), we want to ensure that

$$\frac{p\lambda}{\lambda - (1-p)} \ge 1 + h \cdot w(\lambda) \cdot \exp\big(-l \cdot w(\lambda)\big),$$

which yields, after taking the logarithm on both sides,

$$w(\lambda) \ge \frac{1}{b} \log\left(\frac{1}{1-\lambda}\right) + \frac{1}{b} \log\left(\frac{[\lambda - (1-p)] \cdot h \cdot w(\lambda)}{1-p}\right). \tag{7.4}$$

[17] For simplicity of notation, we assume that $\frac{\lambda+1-p}{2} \cdot w(\lambda)$ is always an integer. This does not change the scaling behavior of $w(\lambda)$.

It is not difficult to verify that for all $p \in (0, 1)$ there exists a constant C such that the above inequality holds for all $\lambda \in (1 - p, 1)$, by letting $w(\lambda) = C \log(\frac{1}{1-\lambda})$. This proves the feasibility of $\sigma^{w(\lambda)}$, which implies that $\pi_{\mathrm{NOB}}^{w(\lambda)}$ is also feasible. This completes the proof of Theorem 3.

8. Concluding remarks and future work. The main objective of this paper is to study the impact of future information on the performance of a class of admissions control problems, with a constraint on the time-average rate of redirection. Our model is motivated as a study of a dynamic resource allocation problem between slow (congestion-prone) and fast (congestion-free) processing resources. It could also serve as a simple canonical model for analyzing delays in large server farms or cloud clusters with resource pooling [20]; cf. Appendix B. Our main results show that the availability of future information can dramatically reduce the delay experienced by admitted customer: the delay converges to a finite constant even as the traffic load approaches the system capacity ("heavy-traffic delay collapse"), if the decision maker is allowed for a sufficiently large lookahead window (Theorem 3).

There are several interesting directions for future exploration. On the theoretical end, a main open question is whether a matching lower-bound on the amount of future information required to achieve the heavy-traffic delay collapse can be proved (Conjecture 1), which, together with the upper bound given in Theorem 3, would imply a duality between delay and the length of lookahead into the future.

Second, we believe that our results can be generalized to the cases where the arrival and service processes are non-Poisson. We note that the π_{NOB} policy is indeed feasible for a wide range of non-Poisson arrival and service processes (e.g., renewal processes), as long as they satisfy a form of strong law of large number, with appropriate time-average rates (Lemma 2). It seems more challenging to generalize results on the optimality of π_{NOB} and the performance guarantees. However, it may be possible to establish a generalization of the delay optimality result using limiting theorems (e.g., diffusion approximations). For instance, with sufficiently well-behaved arrival and service processes, we expect that one can establish a result similar to Proposition 1 by characterizing the resulting queue length process from π_{NOB} as a reflected Brownian motion in $\mathbb{R}_+$, in the limit of $\lambda \to 1$ and $p \to 0$, with appropriate scaling.

Another interesting variation of our problem is the setting where each job comes with a prescribed *size*, or *workload*, and the decision maker is able to observe both the arrival times and workloads of jobs up to a finite lookahead window. It is conceivable that many analogous results can be established for this setting, by studying the associated workload (as opposed to queue length) process, while the analysis may be less clean due to the lack of a simple random-walk-based description of the system dynamics. Moreover, the *server* could potentially exploit additional information of the jobs' workloads in making scheduling decisions, and

it is unclear what the performance and fairness implications are for the design of admissions control policies.

There are other issues that need to be addressed if our offline policies (or policies with a finite lookahead) are to be applied in practice. A most important question can be the impact of *observational noise* to performance, since in reality the future seen in the lookahead window cannot be expected to match the actual realization exactly. We conjecture, based on the analysis of π_{NOB}, that the performance of both π_{NOB}, and its finite-lookahead version, is robust to small noises or perturbations (e.g., if the actual sample path is at most ε away from the predicted one), while it remains to thoroughly verify and quantify the extend of the impact, either empirically or through theory. Also, it is unclear what the best practices should be when the lookahead window is very small relative to the traffic intensity λ ($w \ll \log \frac{1}{1-\lambda}$), and this regime is not covered by the results in this paper (as illustrated in Figure 8).

APPENDIX A: ADDITIONAL PROOFS

A.1. Proof of Lemma 2. Since $\lambda > 1 - p$, with probability one, there exists $T < \infty$ such that the continuous-time queue length process without deletion satisfies $Q^0(t) > 0$ for all $t \geq T$. Therefore, without any deletion, all service tokens are matched with some job after time T. By the stack interpretation, π_{NOB} only deletes jobs that would not have been served, and hence does not change the original matching of service tokens to jobs. This proves the first claim.

By the first claim, since all subsequent service tokens are matched with a job after some time T, there exists some $N < \infty$, such that

$$\widetilde{Q}[n] = \widetilde{Q}[N] + \big(A[n] - A[N]\big) - \big(S[n] - S[N]\big) - I\big(M^{\Psi}, n\big) \tag{A.1}$$

for all $n \geq N$, where $A[n]$ and $S[n]$ are the cumulative numbers of arrival and service tokens by slot n, respectively. The second claim follows by multiplying both sides of equation (A.1) by $\frac{1}{n}$, and using the fact that $\lim_{n\to\infty} \frac{1}{n} A[n] = \frac{\lambda}{\lambda+1-p}$ and $\lim_{n\to\infty} \frac{1}{n} S[n] = \frac{1-p}{\lambda+1-p}$ a.s., $\widetilde{Q}[n] \geq 0$ for all n and $\widetilde{Q}[N] < \infty$ a.s.

A.2. Proof of Lemma 4. (1) Recall the point-wise deletion map, $D_P(Q, n)$, defined in Definition 2. For any initial sample path Q^0, let $Q^1 = D_P(Q^0, m)$ for some $m \in \mathbb{N}$. It is easy to see that, for all $n > m$, $Q^1[n] = Q^0[n] - 1$, if and only if $Q^0[s] \geq 1$ for all $s \in \{m+1, \ldots, n\}$. Repeating this argument $I(M, n)$ times, we have that

$$Q[n] = \widetilde{Q}[n + m_1] = Q^0[n + m_1] - I(M, n + m_1), \tag{A.2}$$

if and only if for all $k \in \{1, \ldots, I(M, n + m_1)\}$,

$$Q^0[s] \geq k \qquad \text{for all } s \in \{m_k + 1, \ldots, n + m_1\}. \tag{A.3}$$

Note that equation (A.3) is implied by (and in fact, equivalent to) the definition of the m_k's (Definition 7), namely, that for all $k \in \mathbb{N}$, $Q^0[s] \geq k$ for all $s \geq m_k + 1$. This proves the first claim.

(2) Suppose $Q[n] = Q[n-1] = 0$. Since $\mathbb{P}(Q^0[t] \neq Q^0[t-1] | Q^0[t-1] > 0) = 1$ for all $t \in \mathbb{N}$ [cf. equation (2.1)] at least one deletion occurs on the slots $\{n-1+m_1, n+m_1\}$. If the deletion occurs on $n+m_1$, we are done. Suppose a deletion occurs on $n-1+m_1$. Then $Q^0[n+m_1] \geq Q^0[n-1+m_1]$, and hence

$$Q^0[n+m_1] = Q^0[n-1+m_1] + 1,$$

which implies that a deletion must also occur on $n+m_1$, for otherwise $Q[n] = Q[n-1]+1 = 1 \neq 0$. This shows that $n = m_i - m_1$ for some $i \in \mathbb{N}$.

Now, suppose that $n = m_i - m_1$ for some $i \in \mathbb{N}$. Let

$$n_k = \inf\{n \in \mathbb{N} : Q^0[n] = k \text{ and } Q^0[t] \geq k,\ \forall t \geq n\}. \tag{A.4}$$

Since the random walk Q^0 is transient, and the magnitude of its step size is at most 1, it follows that $n_k < \infty$ for all $k \in \mathbb{N}$ a.s., and that $m_k = n_k$, $\forall k \in \mathbb{N}$. We have

$$\begin{aligned} Q[n] &\overset{(a)}{=} Q^0[n+m_1] - I(M, n+m_1) \\ &= Q^0[m_i] - I(M, m_i) \\ &\overset{(b)}{=} Q^0[n_i] - i \\ &= 0, \end{aligned} \tag{A.5}$$

where (a) follows from equation (A.2) and (b) from the fact that $n_i = m_i$. To show that $Q[n-1] = 0$, note that since $n = m_i - m_1$, an arrival must have occurred in Q^0 on slot m_i, and hence $Q^0[n-1+m_1] = Q^0[n+m_1] - 1$. Therefore, by the definition of m_i,

$$Q^0[t] - Q^0[n-1+m_1] = (Q^0[t] - Q^0[n+m_1]) + 1 \geq 0 \qquad \forall t \geq n+m_1,$$

which implies that $n-1 = m_{i-1} - m_1$, and hence $Q[n-1] = 0$, in light of equation (A.5). This proves the claim.

(3) For all $n \in \mathbb{Z}_+$, we have

$$\begin{aligned} Q[n] &= Q[m_{I(M,n+m_1)} - m_1] + (Q[n] - Q[m_{I(M,n+m_1)} - m_1]) \\ &\overset{(a)}{=} Q[n] - Q[m_{I(M,n+m_1)} - m_1] \\ &\overset{(b)}{=} Q^0[n+m_1] - Q^0[m_{I(M,n+m_1)}] \\ &\overset{(c)}{=} 0, \end{aligned} \tag{A.6}$$

where (a) follows from the second claim [cf. equation (A.5)], (b) from the fact that there is no deletion on any slot in $\{I(M, n+m_1), \ldots, n+m_1\}$ and (c) from the fact that $n+m_1 \geq I(M, n+m_1)$ and equation (3.3).

A.3. Proof of Lemma 5. Since the random walk X lives in $\mathbb{Z}_+$ and can take jumps of size at most 1, it suffices to verify that

$$\mathbb{P}\Big(X[n+1] = x_1 + 1 | X[n] = x_1, \min_{r \geq n+1} X[r] = 0\Big) = 1 - q$$

for all $x_1 \in \mathbb{Z}_+$. We have

$$\begin{aligned}
&\mathbb{P}\Big(X[n+1] = x_1 + 1 | X[n] = x_1, \min_{r \geq n+1} X[r] = 0\Big) \\
&\quad = \mathbb{P}\Big(X[n+1] = x_1 + 1, \min_{r \geq n+1} X[r] = 0 | X[n] = x_1\Big) \\
&\qquad / \mathbb{P}\Big(\min_{r \geq n+1} X[r] = 0 | X[n] = x_1\Big) \\
&\quad \overset{\text{(a)}}{=} \Big(\mathbb{P}\big(X[n+1] = x_1 + 1 | X[n] = x_1\big) \\
&\qquad \times \mathbb{P}\Big(\min_{r \geq n+1} X[r] = 0 | X[n+1] = x_1 + 1\Big)\Big) \\
&\qquad / \mathbb{P}\Big(\min_{r \geq n+1} X[r] = 0 | X[n] = x_1\Big) \\
&\quad \overset{\text{(b)}}{=} q \cdot \frac{h(x_1 + 1)}{h(x_1)},
\end{aligned} \tag{A.7}$$

where

$$h(x) = \mathbb{P}\Big(\min_{r \geq 2} X[r] = 0 | X[1] = x\Big)$$

and steps (a) and (b) follow from the Markov property and stationarity of X, respectively. The values of $\{h(x) : x \in \mathbb{Z}_+\}$ satisfy the set of harmonic equations

$$h(x) = \begin{cases} q \cdot h(x+1) + (1-q) \cdot h(x-1), & x \geq 1, \\ q \cdot h(1) + 1 - q, & x = 0 \end{cases} \tag{A.8}$$

with the boundary condition

$$\lim_{x \to \infty} h(x) = 0. \tag{A.9}$$

Solving equations (A.8) and (A.9), we obtain the unique solution

$$h(x) = \left(\frac{1-q}{q}\right)^x$$

for all $x \in \mathbb{Z}_+$. By equation (A.7), this implies that

$$\mathbb{P}\Big(X[n+1] = x_1 + 1 | X[n] = x_1, \min_{r \geq n+1} X[r] = 0\Big) = q \cdot \frac{1-q}{q} = 1 - q,$$

which proves the claim.

A.4. Proof of Lemma 8. By the definition of $\overline{F}_{X_1}^{-1}$ and the strong law of large numbers (SLLN), we have

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\big) = \mathbb{E}\big(\mathbb{I}\big(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\big)\big) < \alpha \qquad \text{a.s.} \tag{A.10}$$

Denote by $S_{n,k}$ set of top k elements in $\{X_i : 1 \leq i \leq n\}$. By equation (A.10) and the fact that $H_n \lesssim \alpha n$ a.s., there exists $N > 0$ such that

$$\mathbb{P}\big\{\exists N \text{ s.t. } \min S_{n,H_n} \geq \overline{F}_{X_1}^{-1}(\alpha),\ \forall n \geq N\big\} = 1,$$

which implies that

$$\begin{aligned} &\limsup_{n\to\infty} f\big(\{X_i : 1 \leq i \leq n\}, H_n\big) \\ &\leq \limsup_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} X_i \cdot \mathbb{I}\big(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\big) \qquad (A.11) \\ &= \mathbb{E}\big(X_1 \cdot \mathbb{I}\big(X_1 \geq \overline{F}_{X_1}^{-1}(\alpha)\big)\big) \qquad \text{a.s.}, \end{aligned}$$

where the last equality follows from the SLLN. This proves our claim.

A.5. Proof of Lemma 10. We begin by stating the following fact:

LEMMA 11. *Let $\{X_i : i \in \mathbb{N}\}$ be i.i.d. random variables taking values in $\mathbb{R}_+$, such that for some $a, b > 0$, $\mathbb{P}(X_1 \geq x) \leq a \cdot \exp(-b \cdot x)$ for all $x \geq 0$. Then*

$$\max_{1\leq i\leq n} X_i = o(n) \qquad \text{a.s.}$$

as $n \to \infty$.

PROOF.

$$\begin{aligned} \lim_{n\to\infty} \mathbb{P}\left(\max_{1\leq i\leq n} X_i \leq \frac{2}{b}\ln n\right) &= \lim_{n\to\infty} \mathbb{P}\left(X_1 \leq \frac{2}{b}\ln n\right)^n \\ &\leq \lim_{n\to\infty} \big(1 - a\cdot\exp(-2\ln n)\big)^n \qquad (A.12) \\ &= \lim_{n\to\infty} \left(1 - \frac{a}{n^2}\right)^n \\ &= 1. \end{aligned}$$

In other words, $\max_{1\leq i\leq n} X_i \leq \frac{2}{b}\ln n$ a.s. as $n \to \infty$, which proves the claim. □

Since the $|E_i|$'s are i.i.d. with $\mathbb{E}(|E_1|) = \frac{\lambda+1-p}{\lambda-(1-p)}$ (Proposition 2), we have that, almost surely,

$$m_K^{\Psi} = \sum_{i=0}^{K-1} |E_i| \sim \mathbb{E}\big(|E_1|\big)\cdot K = \frac{\lambda+1-p}{\lambda-(1-p)}\cdot K \qquad \text{as } K \to \infty \tag{A.13}$$

by the strong law of large numbers. By Lemma 11 and equations (6.28), we have

$$\max_{1\le i\le K}|E_i| = o(K) \qquad \text{a.s.} \tag{A.14}$$

as $K\to\infty$. By equation (A.14) and the fact that $I(M^\Psi, m_K^\Psi) = K$, we have

$$\begin{aligned} K - I\big(M^\Psi, l(m_K^\Psi)\big) &= K - I\Big(M^\Psi, m_K^\Psi - \max_{1\le i\le K}|E_i|\Big) \\ &\overset{(a)}{\le} K - I\big(M^\Psi, m_K^\Psi\big) + \max_{1\le i\le K}|E_i| \\ &= \max_{1\le i\le K}|E_i| \\ &= o(K) \qquad \text{a.s.} \end{aligned} \tag{A.15}$$

as $K\to\infty$, where (a) follows from the fact that at most one deletion can occur in a single slot, and hence $I(M, n+m)\le I(M,n)+m$ for all $m,n\in\mathbb{N}$. Since $\widetilde{M}$ is feasible,

$$I(\widetilde{M}, n) \lesssim \frac{p}{\lambda+1-p}\cdot n \tag{A.16}$$

as $n\to\infty$. We have

$$\begin{aligned} h(K) &= \big(K - I\big(M^\Psi, l(m_K^\Psi)\big)\big) + \big(I\big(\widetilde{M}, m_K^\Psi\big) - I\big(M^\Psi, m_K^\Psi\big)\big) \\ &\overset{(a)}{\lesssim} \big(K - I\big(M^\Psi, l(m_K^\Psi)\big)\big) + \frac{p}{\lambda+1-p}\cdot m_K^\Psi - K \\ &\overset{(b)}{\sim} \left(\frac{p}{\lambda+1-p}\cdot\frac{\lambda+1-p}{\lambda-(1-p)} - 1\right)\cdot K, \\ &= \frac{1-\lambda}{\lambda-(1-p)}\cdot K \qquad \text{a.s.} \end{aligned}$$

as $K\to\infty$, where (a) follows from equations (A.13) and (A.16), (b) from equations (A.13) and (A.15), which completes the proof.

APPENDIX B: APPLICATIONS TO RESOURCE POOLING

We discuss in this section some of the implications of our results in the context of a multi-server model for resource pooling [20], illustrated in Figure 2, which has partially motivated our initial inquiry.

We briefly review the model in [20] below, and the reader is referred to the original paper for a more rigorous description. Fix a coefficient $p\in[0,1]$. The system consists of N stations, each of which receives an arrival stream of jobs at rate $\lambda\in(0,1)$ and has one queue to store the unprocessed jobs. The system has a total amount of processing capacity of N jobs per unit time and is divided between

two types of servers. Each queue is equipped with a *local server* of rate $1-p$, which is capable of serving only the jobs directed to the respective station. All stations share a *central server* of rate pN, which always fetches a job from the most loaded station, following a longest-queue-first (LQF) scheduling policy. In other words, a fraction p of the total processing resources is being *pooled* in a centralized fashion, while the remainder is distributed across individual stations. All arrival and service token generation processes are assumed to be Poisson and independent from one another (similarly to Section 2).

A main result of [20] is that even a small amount of resource pooling (small but positive p) can have significant benefits over a fully distributed system ($p=0$). In particular, for any $p>0$, and in the limit as the system size $N\to\infty$, the average delay across the whole system scales as $\sim \log_{1/(1-p)} \frac{1}{1-\lambda}$, as $\lambda\to 1$; note that this is the same scaling as in Theorem 1. This is an exponential improvement over the scaling of $\sim\frac{1}{1-\lambda}$ when no resource pooling is implemented; that is, $p=0$.

We next explain how our problem is connected to the resource pooling model described above, and how the current paper suggests that the results in [20] can be extended in several directions. Consider a similar N-station system as in [20], with the only difference being that instead of the central server fetching jobs from the local stations, the central server simply fetches jobs from a "central queue," which stores jobs redirected from the local stations (see Figure 3). Denote by $\{R_i(t): t\in\mathbb{R}_+\}$, $i\in\{1,\dots,N\}$, the counting process where $R_i(t)$ is the cumulative number of jobs redirected to the central queue from station i by time t. Assume that $\limsup_{t\to\infty}\frac{1}{t}R_i(t)=p-\varepsilon$ almost surely for all $i\in\{1,\dots,N\}$, for some $\varepsilon>0$.[18]

From the perspective of the central queue, it receives an arrival stream R^N, created by merging N redirection streams, $R^N(t)=\sum_{i=1}^N R_i(t)$. The process R^N is of rate $(p-\varepsilon)N$, and it is served by a service token generation process of rate pN. The traffic intensity of the of central queue (arrival rate divided by service rate) is therefore $\rho_c=(p-\varepsilon)N/pN=1-\varepsilon/p<1$. Denote by $Q^N\in\mathbb{Z}_+$ the length of the central queue in steady-state. Suppose that it can be shown that[19]

$$\limsup_{N\to\infty}\mathbb{E}\big(Q^N\big)<\infty. \tag{B.1}$$

A key consequence of equation (B.1) is that, for large values of N, Q^N becomes negligible in the calculation of the system's average queue length: the average

[18]Since the central server runs at rate pN, the rate of $R_i(t)$ cannot exceed p, assuming it is the same across all i.

[19]For an example where this is true, assume that every local station adopts a randomized rule and redirects an incoming job to the central queue with probability $\frac{p-\varepsilon}{\lambda}$ [and that λ is sufficiently close to 1 so that $\frac{p-\varepsilon}{\lambda}\in(0,1)$]. Then $R_i(t)$ is a Poisson process, and by the merging property of Poisson processes, so is $R_N(t)$. This implies that the central queue is essentially an $M/M/1$ queue with traffic intensity $\rho_c=(p-\varepsilon)/p$, and we have that $\mathbb{E}(Q^N)=\frac{\rho_c}{1-\rho_c}$ for all N.

queue length across the whole system coincides with the average queue length among the *local* stations, as $N \to \infty$. In particular, this implies that, in the limit of $N \to \infty$, the task of scheduling for the resource pooling system could alternatively be implemented by running a separate admissions control mechanism, with the rate of redirection equal to $p - \varepsilon$, where all redirected jobs are sent to the central queue, granted that the streams of redirected jobs $(R_i(t))$ are sufficiently well behaved so that equation (B.1) holds. This is essentially the justification for the equivalence between the resource pooling and admissions control problems, discussed at the beginning of this paper (Section 1.2).

With this connection in mind, several implications follow readily from the results in the current paper, two of which are given below:

(1) The original longest-queue-first scheduling policy employed by the central server in [20] is *centralized*: each fetching decision of the central server requires the full knowledge of the queue lengths at all local stations. However, Theorem 1 suggests that the same system-wide delay scaling in the resource pooling scenario could also be achieved by a *distributed* implementation: each server simply runs the same threshold policy, $\pi_{\text{th}}^{L(p-\varepsilon,\lambda)}$, and routes all deleted jobs to the central queue. To prove this rigorously, one needs to establish the validity of equation (B.1), which we will leave as future work.

(2) A fairly tedious stochastic coupling argument was employed in [20] to establish a matching lower bound for the $\sim \log_{1/(1-p)} \frac{1}{1-\lambda}$ delay scaling, by showing that the performance of the LQF policy is no worse than any other online policy. Instead of using stochastic coupling, the lower bound in Theorem 1 immediately implies a lower bound for the resource pooling problem in the limit of $N \to \infty$, if one assumes that the central server adopts a *symmetric* scheduling policy, where the it does not distinguish between two local stations beyond their queue lengths.[20] To see this, note that the rate of $R_i(t)$ are identical under any symmetric scheduling policy, which implies that it must be less than p for all i. Therefore, the lower bound derived for the admissions control problem on a *single queue* with a redirection rate of p automatically carries over to the resource pooling problem. Note that, unlike the previous item, this lower bound does not rely on the validity of equation (B.1).

Both observations above exploit the equivalence of the two problems in the regime of $N \to \infty$. With the same insight, one could also potentially generalize the delay scaling results in [20] to scenarios where the arrival rates to the local stations are nonuniform, or where future information is available. Both extensions seem difficult to accomplish using the original framework of [20], which is based on a fluid model that heavily exploits the symmetry in the system. On the downside, however, the results in this paper tell us very little when system size N is

[20]This is a natural family of policies to study, since all local servers, with the same arrival and service rate, are indeed identical.

small, in which case it is highly conceivable that a centralized scheduling rule, such as the longest-queue-first policy, can out-perform a collection of decentralized admissions control rules.

Acknowledgment. The authors are grateful for the anonymous reviewer's feedback.

REFERENCES

[1] ALTMAN, E. and SHWARTZ, A. (1991). Markov decision problems and state-action frequencies. *SIAM J. Control Optim.* **29** 786–809. MR1111660

[2] AWERBUCH, B., AZAR, Y. and PLOTKIN, S. (1993). Throughput-competitive on-line routing. In *Proceedings of Foundations of Computer Science* (*FOCS*) 32–40. Palo Alto, CA.

[3] AZAR, Y. (1998). On-line load balancing. In *Online Algorithms* (*Schloss Dagstuhl*, 1996). 178–195. Springer, Berlin. MR1676832

[4] BELL, S. L. and WILLIAMS, R. J. (2001). Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy. *Ann. Appl. Probab.* **11** 608–649. MR1865018

[5] BEUTLER, F. J. and ROSS, K. W. (1986). Time-average optimal constrained semi-Markov decision processes. *Adv. in Appl. Probab.* **18** 341–359. MR0840098

[6] BORODIN, A. and EL-YANIV, R. (2005). *Online Computation and Competitive Analysis*, Reissue ed. Cambridge Univ. Press, New York.

[7] CARR, M. and HAJEK, B. (1993). Scheduling with asynchronous service opportunities with applications to multiple satellite systems. *IEEE Trans. Automat. Control* **38** 1820–1833. MR1254317

[8] COFFMAN, E. G. JR., JELENKOVIC, P. and POONEN, B. (1999). Reservation probabilities. *Adv. Perf. Anal.* **2** 129–158.

[9] FISHER, M. and RAMAN, A. (1996). Reducing the cost of demand uncertainty through accurate response to early sales. *Oper. Res.* **44** 87–99.

[10] GALLAGER, R. G. (1996). *Discrete Stochastic Processes*. Kluwer, Boston.

[11] HARRISON, J. M. and LÓPEZ, M. J. (1999). Heavy traffic resource pooling in parallel-server systems. *Queueing Systems Theory Appl.* **33** 339–368. MR1742575

[12] KIM, S. C. and HOROWITZ, I. (2002). Scheduling hospital services: The efficacy of elective surgery quotas. *Omega* **30** 335–346.

[13] LEVI, R. and SHI, C. (2014). Revenue management of reusable resources with advanced reservations. *Oper. Res.* To appear.

[14] LU, Y. and RADOVANOVIĆ, A. (2007). Asymptotic blocking probabilities in loss networks with subexponential demands. *J. Appl. Probab.* **44** 1088–1102. MR2382948

[15] MANDELBAUM, A. and REIMAN, M. I. (1998). On pooling in queueing networks. *Management Science* **44** 971–981.

[16] MANDELBAUM, A. and STOLYAR, A. L. (2004). Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule. *Oper. Res.* **52** 836–855. MR2104141

[17] NAWIJN, W. M. (1990). Look-ahead policies for admission to a single server loss system. *Oper. Res.* **38** 854–862. MR1095946

[18] SMITH, B. L., WILLIAMS, B. M. and OSWALD, R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Cold Spring Harbor Symp. Quant. Biol.* **10** 303–321.

[19] STIDHAM, S. JR. (1985). Optimal control of admission to a queueing system. *IEEE Trans. Automat. Control* **30** 705–713. MR0794203

[20] TSITSIKLIS, J. N. and XU, K. (2012). On the power of (even a little) resource pooling. *Stoch. Syst.* **2** 66. MR2960735

J. SPENCER
COURANT INSTITUTE OF MATHEMATICAL SCIENCES
NEW YORK UNIVERSITY
ROOM 829, 251 MERCER ST.
NEW YORK, NEW YORK 10012
USA
E-MAIL: spencer@courant.nyu.edu

M. SUDAN
MICROSOFT RESEARCH NEW ENGLAND
ONE MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02142
USA
E-MAIL: madhu@mit.edu

K. XU
LABORATORY FOR INFORMATION AND DECISION SYSTEMS (LIDS)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
77 MASSACHUSETTS AVENUE, 32-D666
CAMBRIDGE, MASSACHUSETTS 02139
USA
E-MAIL: kuangxu@mit.edu