QUEUEING WITH FUTURE INFORMATION

By Joel Spencer

Courant Institute of Mathematical Sciences, New York University

By Madhu Sudan

Microsoft Research New England

By Kuang Xu

LIDS, Massachusetts Institute of Technology

Abstract 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. When the future is unknown, the optimal average queue length diverges at rate $\sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$, 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 $\mathcal{O}\left(\log \frac{1}{1-\lambda}\right)$, as $\lambda \to 1$. This leads to the conjecture of an interesting duality between queuing delay and the amount of information about the future.

CONTENTS

1

1. Introduction.

1.1. *"Variable, but Predictable"*. The notion of *queues* have been used extensively as a powerful abstraction in studying dynamic resource allocation problems, where one aims to match *demands* that arrive over time (e.g., manufacturing orders or data queries) with available *resources* (e.g., raw material or computation power). A queue is used to store unprocessed demands. Two important ingredients often make the design and analysis of a queueing system difficult, namely, that 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 non-uniformly distributed across the time horizon. *Unpredictability* means that such non-uniformity "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 variable 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, the adoption of advance booking in the hotel and textile industries has allowed for accurate forecasting of demands ahead of time [12], the availability of monitoring data enables traffic controllers to predict the traffic pattern around potential bottlenecks [4], and advance scheduling for elective surgeries could inform care providers several weeks before the intended appointment [11]. 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. As depicted in Figure 1.1, 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 $(0, 1)$. 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 maximum processing rate, and some form of *admissions control* is necessary to keep the queue from becoming unstable. In particular, upon its arrival to the system, a job will either be *admitted*, in which case it joins the queue, or *redirected*, in which case it 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 delays experienced by the admitted jobs, while obeying the constraint that the average rate at which jobs are
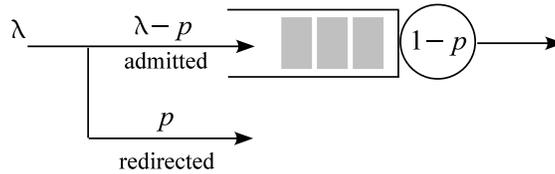
FIGURE 1.1. *An illustration of the admissions control problem, with a constraint on the a rate of redirection.*

redirected *does not exceeded* p. Note that as $\lambda \to 1$, the rate of admitted jobs, which is at least $\lambda - p$, approaches the server capacity $1 - p$, and hence we will refer to the system's behavior when $\lambda \to 1$ as the *heavy-traffic regime*.

One can think of our problem as that of *resource allocation*, where a decision maker tries to match incoming demands with two types of constrained 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 the fast and slow resources are constrained in the sense that their capacities ($1 - p$ and p, respectively) cannot not change over time, by physical or contractual predispositions. The key feature that distinguishes the fast resource from the slow one is the fact that its processing delay is always negligible and does not suffer any short-term congestion effect, so long as the incoming traffic stays below p in the long run. In this interpretation, minimizing the delay among admitted jobs is equivalent to minimizing the average delay across *all* jobs, since the redirected jobs can be thought of 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, in order to absorb any additional incoming traffic that is beyond the maximum capacity of its own in-house server.[1] 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

---

[1] *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.

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 [17], 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 (See Figure 5.1). 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 5.2). This connection will be explored in greater detail in Section 5, where we discuss what the results in this paper tell us about the potential of a resource pooling system.

1.3. *Overview of Main Contributions.* We preview some of the main results below. The formal statements will be given in Section 3.

1.3.1. *Summary of the Problem.* We consider an admissions control problem depicted in Figure 1.1, which is characterized by three parameters: λ, p, and w.

1. Jobs arrives to the system at a rate of λ jobs per unit time, with $\lambda \in 1$. 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, 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.

(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.

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, with

$$C_0^{opt} \sim \log_{\frac{1}{1-p}} \frac{1}{1 - \lambda}, \quad \text{as } \lambda \to 1.$$

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

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

and this limit is achieved by a so-called No-Job-Left-Behind policy.

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

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

This is of practical important, 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, using tools from renewal theory and the theory of random walks. We believe the methodology have the potential to be substantially generalized to incorporate general arrival and service processes, diffusion approximations, as well as observational noises (See Section 9 for a more elaborate discussion).

1.4. *Related Work.* There is an extensive body of literature devoted to the *online* version ($w = 0$) of the admissions (or flow) control problem and its variants; the reader is referred to the survey of [5], 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 $L - 1$ (c.f., [2]). 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 $\log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ that emerges by fixing p and taking $\lambda \to 1$ has not appeared in the literature, to the best our knowledge.

In contrast to our understanding of online admissions control problems, much less is known for settings in which information about the future is taken into consideration. In [6], 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 look-ahead 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, 9] and, more recently, in revenue management [10]. Using simulations, [11] demonstrates that the use of a one- 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 [12] 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 [17] will be discussed in Section 5. For the general topic of resource pooling, interested readers are referred to [13, 14, 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-Left-Behind policy (π_{NOB}), which will be a central object of study for this paper. Section 4 presents two alternative interpretations of the No-Job-Left-Behind policy (as a "stack" and "cave", respectively) that have important structural, as well as algorithmic, implications. Sections 6 through 8 are devoted to the proofs for the results concerning the online, offline and finite-lookahead policies, respectively. Finally, Section 9 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, non-negative integers, and non-negative reals, respectively. We will use the following asymptotic notation throughout: $f(x) \lesssim g(x)$ if $\lim_{x \to 1} \frac{f(x)}{g(x)} \leq 1$, $f(x) \gtrsim g(x)$ if $\lim_{x \to 1} \frac{f(x)}{g(x)} \geq 1$; $f(x) \ll g(x)$ 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 1.1. 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 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.[2] We will 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*. One redirected, a job effectively "disappears", and for this reason, we will use the word **deletion** as a synonymous term for redirection throughout the

---

[2]When the queue is non-empty, 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 [17] 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.

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, [3]

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

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 $n \in \mathbb{Z}_+$,

$$\mathbb{P}\left(Q^0[n+1] = x_2 \mid Q^0[n] = x_1\right) = \begin{cases} \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = -1, \\ 0, & \text{otherwise,} \end{cases} \quad (2.1)$$

if $x_1 > 0$, and

$$\mathbb{P}\left(Q^0[n+1] = x_2 \mid Q^0[n] = x_1\right) = \begin{cases} \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (2.2)$$

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.

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

---

[3]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.

$\Phi(Q)$ be the locations of all arrivals in a discrete-time queue length process Q, i.e.,

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

and for any $M \subset \mathbb{Z}_+$, define the counting process $\{I(M, n) : n \in \mathbb{N}\}$ associated with M as[4]

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

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

   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 Eq. (2.4) is due to the fact that the total rate of events in the system is $\lambda + (1-p)$.[5] 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*

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

---

[4]$|X|$ denotes the cardinality of X.
[5]This is equal to the total rate of jumps in $A(\cdot)$ and $S(\cdot)$.

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

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

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 (Eq. (2.6)), which is at most n, and the limit in Eq. (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 $\tilde{M}$ be two feasible deletion sequences that contain the same elements. Then $D\left(Q^0, M\right) = D\left(Q^0, \tilde{M}\right)$.*

We next define the notion of a deletion policy, which 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 $\mathcal{F}_t = \sigma\left(Q^0(s); s \le t\right)$ be the natural filtration induced by $\left\{Q^0(t) : t \in \mathbb{R}_+\right\}$, and $\mathcal{F}_\infty = \cup_{t \in \mathbb{Z}_+} \mathcal{F}_t$. A w-predictive deletion policy is a mapping, $\pi : \mathbb{Z}_+^{\mathbb{R}_+} \to \mathbb{N}^\infty$, such that*

1. *$M = \pi\left(Q^0\right)$ 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 till 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 $\Pi_w, 0 < w < \infty$, 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 $Q_\pi^\infty = D\left(Q^0, \pi\left(Q^0\right)\right)$, 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 $T_{all} \in \mathbb{R}_+$ the time-average queueing delay experienced by all jobs, where deleted jobs are assumed to have a delay of zero, then

$$\mathbb{E}(T_{all}) = \frac{1}{\lambda} \cdot C(p, \lambda, \pi), \tag{2.10}$$

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

$$\mathbb{E}(T_{adt}) = \frac{1}{\lambda - r_d} \cdot C(p, \lambda, \pi), \tag{2.11}$$

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.

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

$$C^*_{\Pi_w}(p, \lambda) = \inf_{\pi \in \Pi_w} C(p, \lambda, \pi). \tag{2.12}$$

3. Summary of Main Results. We state the main results of this paper in this section, whose proofs will be presented in Sections 6 through 8.

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 7. **(Optimal Online Policies)** *Fix* $p \in (0, 1)$, *and let*

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

Then,

1. $\pi^{L(p,\lambda)}_{th}$ *is feasible for all* $\lambda \in (1 - p, 1)$.
2. $\pi^{L(p,\lambda)}_{th}$ *is asymptotically optimal in* Π_0 *as* $\lambda \to 1$:

$$C\left(p, \lambda, \pi^{L(p,\lambda)}_{th}\right) \sim C^*_{\Pi_0}(p, \lambda) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad as \ \lambda \to 1.$$

PROOF. See Section 6.2. □

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:

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

DEFINITION 8. **(No-Job-Left-Behind Policy**[6]**)** *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*

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

---

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

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

In other words, π_{NOB} would delete a job arriving at time t if and only if the queue length never returns to below the current level in the future, assuming no more deletions are made after t. Note that by the above definition, we have

$$Q^0[n] \geq Q^0[m_i^{\Psi}], \quad \forall n \geq m_i^{\Psi}, \tag{3.3}$$

for all $i \in \mathbb{N}$. Examples of the π_{NOB} policy being applied to a particular sample path is given in Figures 3.1 and 3.2 (illustration), as well as in Figure 4.1 (simulation).
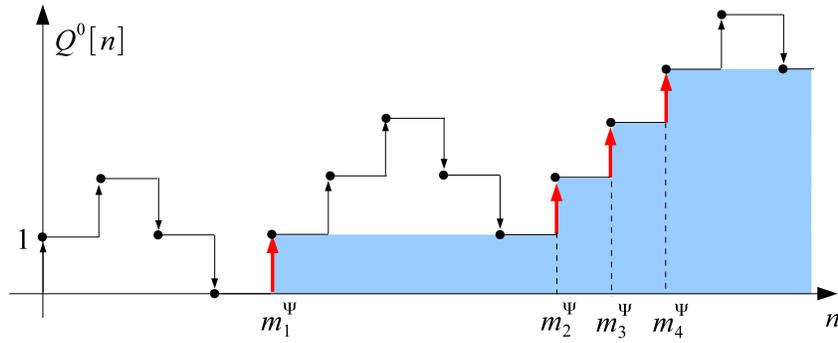
FIGURE 3.1. *Illustration of applying π_{NOB} to an initial sample path, Q^0, where the deletions are marked by bold red arrows.*
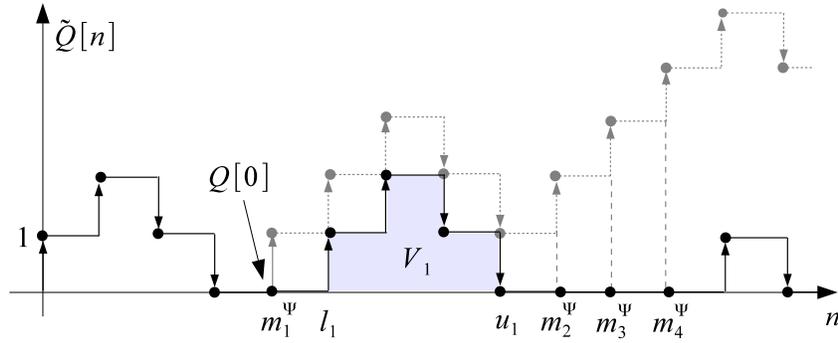

FIGURE 3.2. *The solid lines depict the resulting sample path, $\tilde{Q} = D\left(Q^0, M^{\Psi}\right)$, after applying π_{NOB} to Q^0.*

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.

THEOREM 9.   **(Optimal Offline Policies)** *Fix $p \in (0,1)$.*

1. *π_{NOB} is feasible for all $\lambda \in (1-p,1)$, and* [7]

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

2. *π_{NOB} is asymptotically optimal in Π_∞ as $\lambda \to 1$:*

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

PROOF. See Section 7.4.   □

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 $\sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ for the online policies, given by Theorem 7 (See Figure 3.3 for an illustration of this difference). A "cave" interpretation of the No-Job-Left-Behind policy, to be introduced in Section 4.2, will help us understand intuitively *why* such a drastic discrepancy exists between the online and offline heavy-traffic scaling behaviors. See discussion in Section 4.2.1.

Also, as a by-product of Theorem 9, observe that

$$\lim_{\lambda \to 1} C^*_{\Pi_\infty}(p,\lambda) \sim \frac{1}{p},$$

as rate of deletion $p \to 0$. This is intuitively consistent with our concept of flexibility: that delay should degenerate as the system's ability to redirect 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

---

[7]It is easy to see that π_{NOB} is not a very efficient deletion policy for relative small values of λ. In fact, $C(p,\lambda,\pi_{NOB})$ 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 $1-\lambda-\epsilon$, so that the total rate of arrival is $1-\epsilon$, where $\epsilon \approx 0$. This reasoning implies that $(1-p)/p$ is a uniform upper-bound of $C^*_{\Pi_\infty}(p,\lambda)$ for all $\lambda \in (0,1)$.

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.
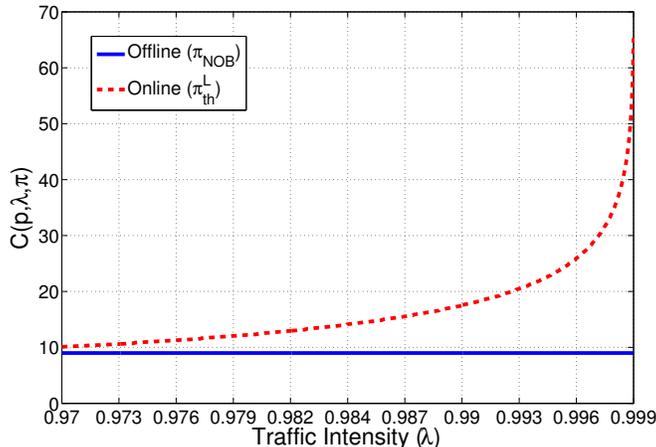
FIGURE 3.3. *Comparison between the heavy-traffic delay scalings associated with π_{NOB} and $\pi_{th}^{L(p,\lambda)}$, with $p = 0.1$ and $\lambda \to 1$.*

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 $\lambda \to 1$. 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.

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

$$W(n) = \sup\{k \in \mathbb{Z}_+ : T_{n+k} \leq T_n + w\}. \tag{3.5}$$

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

$$U(Q, n, x) = \inf\{j \in \{1, \ldots, x\} : Q[n+j] = Q[n] - 1\}. \tag{3.6}$$

DEFINITION 10.   *(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_{NOB}^w, deletes all arrivals in the set Ψ^w, where*

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

It is easy to see that π_{NOB}^w is simply π_{NOB} applied within the confinement
of a finite window: a job at t is deleted if and only if the queue length 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 $C\left(p, \lambda, \pi_{NOB}^w\right) \leq C\left(p, \lambda, \pi_{NOB}\right)$ 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. Note that this result implies Theorem 9 and is hence stronger.

THEOREM 11.   **(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 $\pi_{NOB}^{w(\lambda)}$ is feasible, and

$$C\left(p, \lambda, \pi_{NOB}^{w(\lambda)}\right) \leq C\left(p, \lambda, \pi_{NOB}\right) = \frac{1-p}{\lambda - (1-p)}, \qquad (3.7)$$

*Since $C_{\Pi_{w(\lambda)}}^*(p, \lambda) \geq C_{\Pi_\infty}^*(p, \lambda)$ and $C_{\Pi_{w(\lambda)}}^*(p, \lambda) \leq C\left(p, \lambda, \pi_{NOB}^{w(\lambda)}\right)$, we also
have that*

$$\lim_{\lambda \to 1} C_{\Pi_{w(\lambda)}}^*(p, \lambda) = \lim_{\lambda \to 1} C_{\Pi_\infty}^*(p, \lambda) = \frac{1-p}{p}. \qquad (3.8)$$

PROOF. See Section 8.1. $\square$

3.3.1. *Delay-Information Duality.*   Theorem 11 says that one can attain
the same heavy-traffic delay performance as the the optimal offline algo-
rithm, 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 con-
jecture that this is the case, in the sense that thee 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 $w(\lambda) = \Theta\left(\log \frac{1}{1-\lambda}\right)$.

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 $w(\lambda) = \Theta\left(\log \frac{1}{1-\lambda}\right)$. It would also imply the existence of a symmetric dual relationship between *future information* and *queueing delay*: $\Theta\left(\log \frac{1}{1-\lambda}\right)$ amount of information is required to achieve a finite delay limit, and one has to suffer $\Theta\left(\log \frac{1}{1-\lambda}\right)$ in delay, if only finite amount of future information is available.
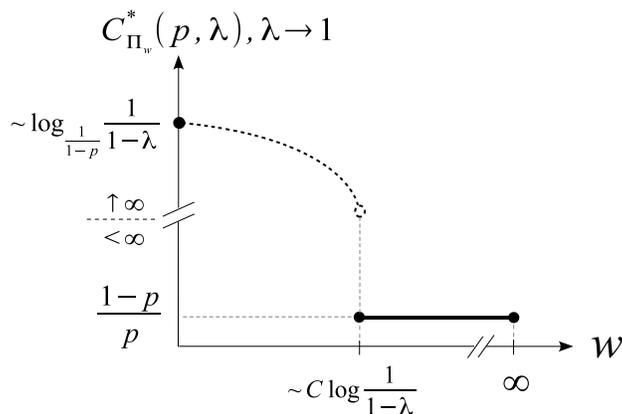
FIGURE 3.4. *"Delay v.s Information." Best achievable heavy traffic delay scaling as a function of the size of the lookahead window, w. Results presented in this paper are illustrated in the solid lines and circles, and the gray dotted line depicts our conjecture of the unknown regime of $0 < w(\lambda) \lesssim \log\left(\frac{1}{1-\lambda}\right)$.*

Figure 3.4 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 end point reflects the view of Conjecture 1.

4. Interpretations of π_{NOB}. We present two equivalent ways of describing the No-Job-Left-Behind policy π_{NOB}. While the interpretations may be interesting in their own right, they also provide us with operational insights into the dynamics of the policy. In particular, the *stack interpretation* helps us derive asymptotic deletion rate of π_{NOB} in a simple manner,

and the *cave interpretation*, which takes a time-reversal point of view, 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 3.1 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 8 is equivalent to deleting all jobs that are labeled "left-behind", hence the namesake "No-Job-Left-Behind". Figure 3.2 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 $Q = D\left(Q^0, M^\Psi\right)$. We have*

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

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

PROOF. See Appendix A.1 □

4.2. *Cave Interpretation.* We now view the sample path of $\left\{Q^0[n] : n \in \mathbb{N}\right\}$ as the wall of a cave: the x axis is the "floor", the area above Q^0 the "rock", and the cave opens up towards right. Now, suppose there is a light source placed at $n = \infty$, emitting parallel beams of light (illustrated by the blue shades in Figure 3.1) into the cave from the right. By Definition 8, it is easy to see that the deletions made by π_{NOB} are precisely the areas on the wall that are "lit" by this light source.

The cave interpretation shows that the deletions made by π_{NOB} are arguably more natural when viewing the process Q^0 *in reverse*. It may be counter-intuitive that the notion of time should matter, since the problem is "offline" after all. However, as we will see in the next section, the time-reverse view leads naturally to an algorithm of computing M^Ψ over a finite time horizon $n \in \{1, \ldots, N\}$, whose running time scales linearly with respect to N as $N \to \infty$, and is very simple to describe.

4.2.1. *"Anticipation" v.s "Reaction".* Another important value of the cave interpretation is that it demonstrates the power of π_{NOB}'s being highly *anticipatory*, in a geometrically intuitive manner. Looking at Figure 3.1, one sees immediately that the wall areas "under light" correspond to all the segments where the initial sample path Q^0 are taking a consecutive "upward hike". In other words, the policy π_{NOB} begins to delete jobs precisely when it anticipates that the arrivals are *just about to* get intense. Similarly, a wall area will be "in the shade" only if the wall 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, $\pi_{th}^{L(p,\lambda)}$, 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 4.1. 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 $\pi_{th}^{L(p,\lambda)}$ during this period.

To summarize this comparison with 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 9), while the delay under even the best online policy diverges to infinity as $\lambda \to 1$ (Theorem 7).

4.2.2. *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
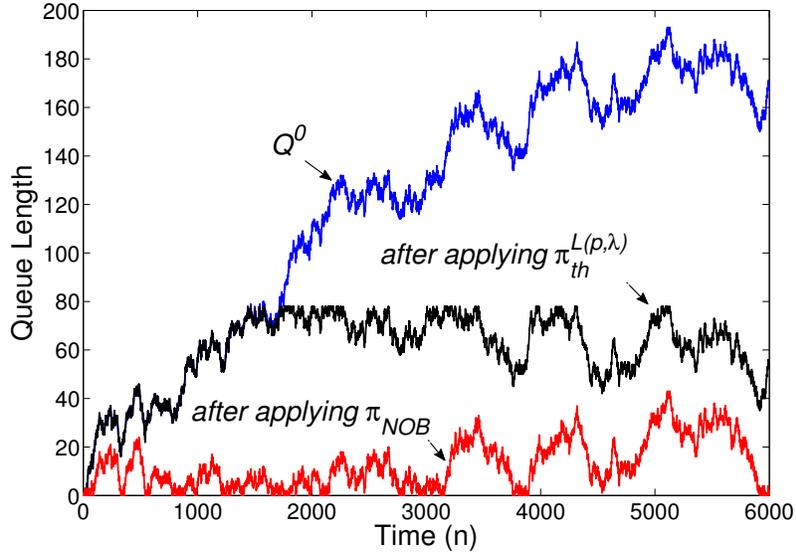
FIGURE 4.1. *Example sample paths of Q^0 and those obtained after applying $\pi_{th}^{L(p,\lambda)}$ and π_{NOB} to Q^0, with $p = 0.05$ and $\lambda = 0.999$.*

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" is achieved in Q^0, when viewed in the reverse order of time. Note that these are precisely the slots "under light" according to the cave interpretation (Section 4.2).

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. 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 [17], illustrated in Figure 5.1, which has partially motivated our initial inquiry.

We briefly review the model in [17] 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 (same as in Section 2).

A main result of [17] 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 $\sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$, as $\lambda \to 1$ (note this is the same scaling as in Theorem 7). This is an exponential improvement over the scaling of $\sim \frac{1}{1-\lambda}$ when no resource pooling is implemented, i.e., $p = 0$.

We next explain how our problem is intimately connected to the resource pooling model described above, and how the current paper suggests that the
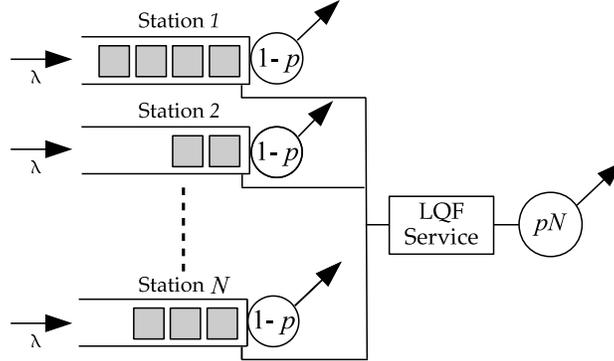
FIGURE 5.1. *Illustration of a model for resource pooling with distributed and centralized resources, [17].*
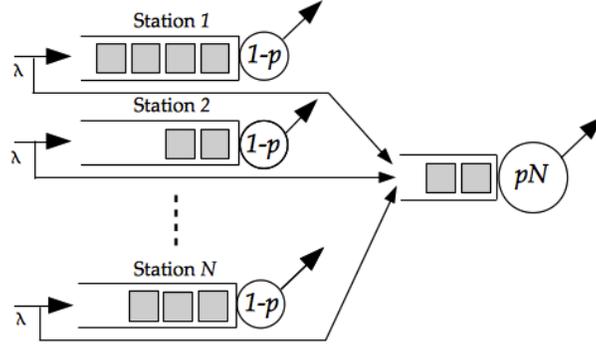
FIGURE 5.2. *Resource pooling using a central queue.*

results in [17] can be extended along several directions. Consider a similar N-station system as in [17], 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 5.2. Denote by $\{R_i(t) : t \in \mathbb{R}_+\}$, $i \in \{1, \ldots, N\}$, 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 $\limsup_{t \to \infty} \frac{1}{t} R_i(t) = p - \epsilon$ almost surely for all $i \in \{1, \ldots, N\}$, for some $\epsilon > 0$.[8]

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

_____

[8]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.

process R^N is of rate $(p-\epsilon)N$, 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 $\rho_c = (p-\epsilon)N/pN = 1 - \epsilon/p < 1$. Denote by $Q^N \in \mathbb{Z}_+$ the length of the central queue in steady-state. Suppose that it can be shown that[9]

$$\limsup_{N\to\infty} \mathbb{E}\left(Q^N\right) < \infty. \tag{5.1}$$

A key consequence of Eq. (5.1) is that, for large values of N, Q^N becomes negligible in the calculation of the system's average queue length: the average 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 $p - \epsilon$, 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 Eq. (5.1) 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 follows readily from the results in the current paper, two of which are given below

1. The original LQF scheduling policy employed by the central server in [17] is *centralized*: each fetching decision of the central server requires the full knowledge of the queue lengths at all local stations. However, Theorem 7 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, $\pi_{th}^{L(p-\epsilon,\lambda)}$, and routes all deleted jobs to the central queue. To prove this rigorously, one needs to establish the validity of Eq. (5.1), which we will leave as future work.

2. A fairly tedious stochastic coupling argument was employed in [17] to establish a matching lower bound for the $\sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ 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

---

[9]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 $\frac{p-\epsilon}{\lambda}$ (and that λ is sufficiently close to 1 so that $\frac{p-\epsilon}{\lambda} \in (0,1)$). 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 $\rho_c = (p-\epsilon)/p$, and we have that $\mathbb{E}\left(Q^N\right) = \frac{\rho_c}{1-\rho_c}$ for all N.

bound in Theorem 7 immediately implies a lower bound for the re-
source 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.[10] 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 automat-
ically carries over to the resource pooling problem. Note that, unlike
the previous item, this lower bound does not rely on the validity of
Eq. (5.1).

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 [17] to scenarios where the arrival
rates to the local stations are non-uniform, or where future information is
available. Both extensions seem difficult to accomplish using the original
framework of [17], 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 *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.

6. Optimal Online Policies. Starting from this section and through
Section 8, we present the proofs of the results stated in Section 3.

We begin with showing Theorem 7, 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.

6.1. *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

---

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

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 interval, 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,

$$R(Q_k, a_k, \tau_k) = -Q_k \cdot \tau_k, \tag{6.1}$$
$$K(Q_k, a_k, \tau_k) = \lambda(1 - a_k)\tau_k, \tag{6.2}$$

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

$$\bar{R}_\pi = \liminf_{n \to \infty} \frac{\mathbb{E}_\pi \left(\sum_{k=1}^n R(Q_k, a_k, \tau_k) \right)}{\mathbb{E}_\pi \left(\sum_{k=1}^n \tau_k \right)} \tag{6.3}$$

while obeying the average-cost constraint

$$\bar{C}_\pi = \limsup_{n \to \infty} \frac{\mathbb{E}_\pi \left(\sum_{k=1}^n K(Q_k, a_k, \tau_k) \right)}{\mathbb{E}_\pi \left(\sum_{k=1}^n \tau_k \right)} \leq p. \tag{6.4}$$

To see why this MDP solves our deletion problem, observe that $\bar{R}_\pi$ is the negative of the time-average queue length, and $\bar{C}_\pi$ 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 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[11], 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 [2]), which ours do satisfy (Eqs. (6.1) and (6.2)). Theorem 7 will be proved using the next known result (adapted from Theorem 4.4 in [2]):

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 $\xi \in [0,1]$.

---

[11]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.

6.2. *Proof of Theorem 7.*

PROOF. **(Theorem 7)** 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 (Eq. (6.4)). 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 6.1, 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

$$C(p, \lambda, \pi_{th}^{L^*-1}) \leq C(p, \lambda, \pi_p^*) \leq C(p, \lambda, \pi_{th}^{L^*}), \tag{6.5}$$

where L^* is defined as in Lemma 3, and $C(p, \lambda, \pi)$ is the time-average queue length under policy π, defined in Eq. (2.9).
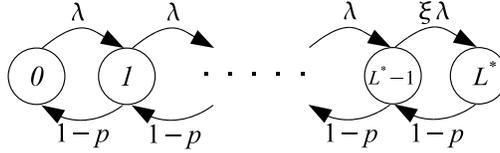
FIGURE 6.1. *The truncated birth-death process induced by π_p^*.*

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

$$\mu_i^L = \left(\frac{\lambda}{1-p}\right)^i \cdot \left(\frac{1 - \frac{\lambda}{1-p}}{1 - \left(\frac{\lambda}{1-p}\right)^{L+1}}\right), \quad \forall 1 \leq i \leq L, \tag{6.6}$$

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

$$C(p, \lambda, \pi_{th}^L) = \sum_{i=1}^{L} i \cdot \mu_i^L$$

$$= \frac{\theta}{(\theta - 1)(\theta^{L+1} - 1)} \cdot \left[(2\theta - 1)(\theta^L - 1) + L(\theta - 1)\theta^L\right], \tag{6.7}$$

where $\theta = \frac{\lambda}{1-p}$. Note that when $\lambda > 1 - p$, μ_i^L is decreasing with respect to L for all $i \in \{0, 1, \dots, L\}$ (Eq. (6.6)), which implies that the time-average queue length is monotonically increasing in L, i.e.,

$$
\begin{aligned}
&C(p, \lambda, \pi_{th}^{L+1}) - C(p, \lambda, \pi_{th}^L) \\
=&(L+1) \cdot \mu_{L+1}^{L+1} + \sum_{i=0}^{L} i \cdot (\mu_i^{L+1} - \mu_i^L) \\
\geq&(L+1) \cdot \mu_{L+1}^{L+1} + L \cdot \left(\sum_{i=0}^{L} \mu_i^{L+1} - \mu_i^L \right) \\
=&(L+1) \cdot \mu_{L+1}^{L+1} + L \cdot \left(1 - \mu_i^{L+1} - 1 \right) \\
=&\mu_{L+1}^{L+1} > 0.
\end{aligned}
\tag{6.8}
$$

It is also easy to see that, whenever $\theta > 1$,

$$
C(p, \lambda, \pi_{th}^L) \sim \frac{L\theta^{L+1}}{\theta^{L+1} - 1} \sim L, \quad \text{as } L \to \infty.
\tag{6.9}
$$

Since deletions only occur when $Q(t)$ is in state L, from Eq. (6.6), the average rate of deletions in continuous time under π_{th}^L is given by,

$$
r_d \left(p, \lambda, \pi_{th}^L, \right) = \lambda \cdot \pi_L = \lambda \cdot \left(\frac{\lambda}{1-p} \right)^L \cdot \left(\frac{1 - \frac{\lambda}{1-p}}{1 - \left(\frac{\lambda}{1-p} \right)^{L+1}} \right).
\tag{6.10}
$$

Define

$$
L(x, \lambda) = \min \left\{ L \in \mathbb{Z}_+ : r_d \left(p, \lambda, \pi_{th}^L, \right) \leq x \right\},
\tag{6.11}
$$

that is, $L(x, \lambda)$ is the smallest L for which π_{th}^L remains feasible, given an deletion rate constraint of x. Using Eqs. (6.10) and (6.11) to solve for $L(p, \lambda)$, we obtain, after some algebra,

$$
L(p, \lambda) = \left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil \sim \log_{\frac{1}{1-\tilde{p}}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1,
\tag{6.12}
$$

and, by combining Eq. (6.12) and Eq. (6.9) with $L = L(p, \lambda)$, we have

$$
C(p, \lambda, \pi_{th}^{L(p,\lambda)}) \sim L(p, \lambda) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1.
\tag{6.13}
$$

By Eqs. (6.8) and (6.11), we know that $\pi_{th}^{L(p,\lambda)}$ achieves the minimum average queue length among all feasible threshold policies. By Eq. (6.5), we must have that

$$
C \left(p, \lambda, \pi_{th}^{L(p,\lambda)-1} \right) \leq C(p, \lambda, \pi_p^*) \leq C \left(p, \lambda, \pi_{th}^{L(p,\lambda)} \right),
\tag{6.14}
$$

Since Lemma 3 only applies when $B < \infty$, Eq. (6.14) holds whenever the buffer size, B, is greater than $L(p, \lambda)$ but finite. We next extend Eq. (6.14) 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 (Eq. (6.4)). The upper bound on $C(p, \lambda, \pi_p^*)$ in Eq. (6.14) automatically holds for $C(p, \lambda, \nu_p^*)$, since $C(p, \lambda, \pi_{th}^{L(p,\lambda)})$ is still feasible when $B = \infty$. It remains to show a lower bound of the form

$$C(p, \lambda, \nu_p^*) \geq C\left(p, \lambda, \pi_{th}^{L(p,\lambda)-2}\right) \tag{6.15}$$

when $B = \infty$, which, together with the upper bound, will have implied that the scaling of $C(p, \lambda, \pi_{th}^{L(p,\lambda)})$ (Eq. (6.13)) carries over to ν_p^*,

$$C\left(p, \lambda, \nu_p^*\right) \sim C(p, \lambda, \pi_{th}^{L(p,\lambda)}) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1, \tag{6.16}$$

thus proving Theorem 7.

To show Eq. (6.15), 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

$$b_q^*(B') = \mathbb{I}\left(q \leq B'\right) \cdot b_q^*,$$

for all $q \geq 0$. 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

$$\lim_{B' \to \infty} \mu_i^*(B') = \mu_i^*, \tag{6.17}$$

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

$$\lim_{B' \to \infty} C\left(p, \lambda, \nu_p^*(B')\right) = C\left(p, \lambda, \nu_p^*\right). \tag{6.18}$$

By Eq.(6.17) and the fact that $b_i^*(B') = b_i^*$ for all $0 \leq i \leq B'$, we have that[12]

$$\lim_{B' \to \infty} r_d\left(p, \lambda, \nu_p^*(B')\right) = \lim_{B' \to \infty} \lambda \sum_{i=0}^{\infty} \mu_i^*(B') \cdot \left(1 - b_i^*(B')\right) = r_d\left(p, \lambda, \nu_p^*\right) \leq p. \tag{6.19}$$

---

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

It is not difficult to verify, from the definition of $L(p, \lambda)$ (Eq. (6.11)), that

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

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

$$C\left(p, \lambda, \nu_p^*(B')\right) \leq C\left(p, \lambda, \nu_p^*\right) + \delta, \tag{6.20}$$

$$L\left(\lambda, r_d\left(p, \lambda, \nu_p^*(B')\right)\right) \geq L(p, \lambda) - 1, \tag{6.21}$$

Let $p' = r_d\left(p, \lambda, \nu_p^*(B')\right)$. Since $b_i^*(B') = 0$ for all $i \geq B' + 1$, by Eq. (6.21) we have

$$C\left(p, \lambda, \nu_p^*(B')\right) \geq C\left(p, \lambda, \pi_{p'}^*\right), \tag{6.22}$$

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

$$
\begin{aligned}
& C\left(p, \lambda, \nu_p^*\right) + \delta \\
& \overset{(a)}{\geq} C\left(p, \lambda, \nu_p^*(B')\right) \\
& \overset{(b)}{\geq} C\left(p, \lambda, \pi_{p'}^*\right) \\
& \overset{(c)}{\geq} C\left(p, \lambda, \pi_{th}^{L(p', \lambda) - 1}\right) \\
& \overset{(d)}{\geq} C\left(p, \lambda, \pi_{th}^{L(p, \lambda) - 2}\right),
\end{aligned}
\tag{6.23}
$$

where the inequalities (a) through (d) follow from Eqs. (6.20), (6.22), (6.14), and (6.21), respectively. Since Eq. (6.23) holds for all $\delta > 0$, we have proven Eq. (6.15). This completes the proof of Theorem 7. $\qquad \square$

7. Optimal Offline Policies. The proof of Theorem 9 is given in this section, in two parts. In the first part (Section 7.2), 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, $C(p, \lambda, \pi_{NOB})$, 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 7.3) focuses on showing the heavy-traffic optimality of π_{NOB} among the class of all feasible offline policies, namely, that $\lim_{\lambda \to 1} C(p, \lambda, \pi_{NOB}) = \lim_{\lambda \to 1} C_{\Pi_\infty}^*(p, \lambda)$, which, together with the first part, proves Theorem 9. 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 optimal greedy deletion rule, which is optimal over any a finite time horizon, N. We then show that the discrepancy between the two, in terms of the time-average queue length, diminishes as $N \to \infty$ and $\lambda \to 1$ (with the two limits taken in this order).

7.1. *Additional Notation.*   Define $\tilde{Q}$ as the resulting queue length process after applying π_{NOB}

$$\tilde{Q} = D\left(Q^0, M^\Psi\right).$$

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

$$Q[n] = \tilde{Q}[n + m_1^\Psi], \quad n \in \mathbb{Z}_+. \tag{7.1}$$

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

$$Q[l-1] = Q[u] = 0, \text{ and } Q[n] > 0 \text{ for all } n \in \{l, \ldots, u-1\}. \tag{7.2}$$

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 3.2.

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

$$E_i = \left\{m_i^\Psi - m_1^\Psi, m_i^\Psi + 1 - m_1^\Psi, \ldots, m_{i+1}^\Psi - 1 - m_1^\Psi\right\}, \tag{7.3}$$

as the ith **deletion epoch**.

7.2. *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*

$$Q[n] = Q[n-1] = 0, \tag{7.4}$$

   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 $x_1, x_2 \in \mathbb{Z}_+$,

$$\mathbb{P}\left(Q[n+1] = x_2 \mid Q[n] = x_2\right) = \begin{cases} \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = -1, \\ 0, & otherwise, \end{cases}$$

if $x_1 > 0$, and

$$\mathbb{P}\left(Q[n+1] = x_2 \mid Q[n] = x_1\right) = \begin{cases} \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = 0, \\ 0, & otherwise, \end{cases}$$

if $x_1 = 0$.

PROOF. For a sequence $\{X[n] : n \in \mathbb{N}\}$ and $s, t \in \mathbb{N}$, $s \leq t$, we will use the short-hand
$$X_s^t = \{X[s], \ldots, X[t]\}.$$
Fix $n \in N$, and a sequence $(q_1, \ldots, q_n) \subset \mathbb{Z}_+^n$. We have

$$\mathbb{P}\left(Q[n] = q[n] \,\middle|\, Q_1^{n-1} = q_1^{n-1}\right)$$

$$= \sum_{k=1}^{n} \sum_{\substack{t_1, \ldots, t_k, \\ t_k \leq n-1+t_1}} \mathbb{P}\left(Q[n] = q[n] \,\middle|\, Q_1^{n-1} = q_1^{n-1}, m_1^k = t_1^k, m_{k+1} \geq n + t_1\right)$$

$$\cdot \mathbb{P}\left(m_1^k = t_1^k, m_{k+1} \geq n + t_1 \,\middle|\, Q_1^{n-1} = q_1^{n-1}\right) \tag{7.5}$$

Restricting to the values of t_i's and $q[i]$'s under which the summand is

non-zero, the first factor in the summand can be written as

$$
\mathbb{P}\left(Q[n] = q[n] \,\middle|\, Q_1^{n-1} = q_1^{n-1}, m_1^k = t_1^k, m_{k+1} \geq n + t_1\right)
$$

$$
= \mathbb{P}\left(\tilde{Q}[n + m_1] = q[n] \,\middle|\, \tilde{Q}_{m_1+1}^{m_1+n-1} = q_1^{n-1}, m_1^k = t_1^k, m_{k+1} \geq n + t_1\right)
$$

$$
\stackrel{(a)}{=} \mathbb{P}\left(Q^0[n + t_1] = q[n] + k \,\middle|\, Q^0[s + t_1] = q[s] + I\left(\{t_i\}_{i=1}^k, s + t_1\right), \forall 1 \leq s \leq n - 1,\right.
$$

$$
\left. \text{and} \ \min_{r \geq n + t_1} Q^0[r] \geq k\right)
$$

$$
\stackrel{(b)}{=} \mathbb{P}\left(Q^0[n + t_1] = q[n] + k \,\middle|\, Q^0[n - 1 + t_1] = q[n - 1] + k, \ \text{and} \ \min_{r \geq n + t_1} Q^0[r] \geq k\right),
$$
$$
\tag{7.6}
$$

where $\tilde{Q}$ was defined in Eq. (7.1). 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 $\left\{\min_{r \geq n + t_1} Q^0[r] \geq k\right\}$, $\left\{Q^0[n + t_1] = q[n] + k\right\}$, and their intersection, depend only on the values of $\left\{Q^0[s] : s \geq n + t_1\right\}$, and are hence independent of $\left\{Q^0[s] : 1 \leq s \leq n - 2 + t_1\right\}$ 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

$$
\mathbb{P}\left(Q^0[n + t_1] = q[n - 1] + 1 + k \,\middle|\, Q^0[n - 1 + t_1] = q[n - 1] + k,\right.
$$

$$
\left. \text{and} \ \min_{r \geq n + t_1} Q^0[r] \geq k\right)
$$

$$
= \frac{1 - p}{\lambda + 1 - p}, \tag{7.7}
$$

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

$$
n < m_{k+1} - 1 + m_1. \tag{7.8}
$$

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

$$
q[n] > 0 \ \text{implies} \ Q^0[t] = k, \ \text{for some} \ t \geq n + 1 + m_1. \tag{7.9}
$$

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 Eq. (7.9), we have that

$$
q[n - 1] > 0 \ \text{implies} \ Q^0[t] = k, \ \text{for some} \ t \geq n + m_1. \tag{7.10}
$$

It is important to note that, despite the similarity in conclusions, Eqs. (7.9) and (7.10) are different in their assumptions (i.e., $q[n]$ versus $q[n-1]$). We have

$$\mathbb{P}\Big(Q^0[n+t_1] = q[n-1] + 1 + k \Big| Q^0[n-1+t_1] = q[n-1] + k,$$
$$\text{and} \min_{r \geq n+t_1} Q^0[r] \geq k\Big)$$

$$\overset{(a)}{=} \mathbb{P}\Big(Q^0[n+t_1] = q[n-1] + 1 + k \Big| Q^0[n-1+t_1] = q[n-1] + k,$$
$$\text{and} \min_{r \geq n+t_1} Q^0[r] = k\Big)$$

$$\overset{(b)}{=} \mathbb{P}\Big(Q^0[2] = q[n-1] + 1 \Big| Q^0[1] = q[n-1], \text{ and } \min_{r \geq 2} Q^0[r] = 0\Big)$$

$$\overset{(c)}{=} \frac{1-p}{\lambda + 1 - p}, \tag{7.11}$$

where (a) follows from Eq. (7.10), (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* $n \in \mathbb{N}$,

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

if $x_1 > 0$, *and*

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

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

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

if $x_1 > 0$, *and*

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

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

$$
\mathbb{P}\Big(Q^0[n+t_1] = q[n-1] + 1 + k \,\Big|\, Q^0[n-1+t_1] = q[n-1] + k,
$$

$$
\text{and } \min_{r \ge n+t_1} Q^0[r] \ge k\Big)
$$

$$
\overset{(a)}{=} \mathbb{P}\Big(Q^0[n+t_1] = 1 + k, \text{ and } \min_{r > n+t_1} Q^0[r] = k \,\Big|\, Q^0[n-1+t_1] = k,
$$

$$
\text{and } \min_{r \ge n+t_1} Q^0[r] \ge k\Big)
$$

$$
\overset{(b)}{=} \mathbb{P}\Big(Q^0[2] = 2, \text{ and } \min_{r > 2} Q^0[r] = 1 \,\Big|\, Q^0[1] = 1, \text{ and } \min_{r \ge 2} Q^0[r] \ge 1\Big),
$$

$$
\overset{\triangle}{=} x, \tag{7.12}
$$

where (a) follows from Eq. (7.9) (note its difference with Eq. (7.10)), and (b) from the stationarity and space-homogeneity of Q^0, and the assumption that $k \ge 1$ (Eq. (7.5)).

Since Eqs. (7.11) and (7.12) hold for all $x_1, k \in \mathbb{Z}_+$ and $n \ge m_1 + 1$, by Eq. (7.5), we have that

$$
\mathbb{P}\Big(Q[n] = q[n] \,\Big|\, Q_1^{n-1} = q_1^{n-1}\Big) = \begin{cases} \frac{1-p}{\lambda+1-p}, & q[n] - q[n-1] = 1, \\ \frac{\lambda}{\lambda+1-p}, & q[n] - q[n-1] = -1, \\ 0, & \text{otherwise,} \end{cases} \tag{7.13}
$$

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

$$
\mathbb{P}\Big(Q[n] = q[n] \,\Big|\, Q_1^{n-1} = q_1^{n-1}\Big) = \begin{cases} x, & q[n] - q[n-1] = 1, \\ 1-x, & q[n] - q[n-1] = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{7.14}
$$

if $q[n-1] = 0$, where x represents the value of the probability in Eq. (7.12). 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 our claim.

To do so, one can in principle directly calculate the probability in line (b) of Eq. (7.12), which can be quite difficult to do. We will use an indirect

approach that is computationally much simpler. As a by-product, we will also get a better understanding of an important regenerative structure of π_{NOB}, which will be useful in subsequent analysis. By Eqs. (7.13) and (7.14), 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,

$$\pi_i = \pi_0 \frac{x(\lambda + 1 - p)}{\lambda} \cdot \left(\frac{1-p}{\lambda}\right)^{i-1}, \quad \forall i \geq 1, \tag{7.15}$$

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

$$\pi_0 = \frac{1}{1 + x \cdot \frac{\lambda + 1 - p}{\lambda - (1-p)}}. \tag{7.16}$$

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

$$\lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}(Q[t] = 0) = \pi_0 = \frac{1}{1 + x \cdot \frac{\lambda + 1 - p}{\lambda - (1-p)}}. \tag{7.17}$$

Next, we would like to know many of these visits to the state zero correspond to a deletion.

Recall the notion of a busy period and deletion epoch, defined in Eqs. (7.2) and (7.3), 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, i.e., $m_i + 1 = m_{i+1}$. 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 in Q consists of the slot $m_i - m_1$, plus N_i busy periods, where the N_i's are i.i.d, with[13]

$$N_1 \overset{d}{=} \mathrm{Geo}(1 - x) - 1, \tag{7.18}$$

and hence

$$|E_i| = 1 + \sum_{j=1}^{N_i} B_{i,j}, \tag{7.19}$$

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.

---

[13] $\mathrm{Geo}(p)$ 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 the state $(0,0)$, and hence are i.i.d. Using the Elementary Renewal Theorem, we have

$$\lim_{n \to \infty} \frac{1}{n} I(M, n) = \frac{1}{\mathbb{E}(|E_1|)}, \quad a.s., \tag{7.20}$$

and by treating 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 zero 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 (c.f., Theorem 6, Chapter 3, [3]), that

$$\lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}(Q[t] = 0) = \frac{\mathbb{E}(R_1)}{\mathbb{E}(|E_1|)} = \frac{\mathbb{E}(N_1) + 1}{\mathbb{E}(|E_1|)}, \quad a.s., \tag{7.21}$$

by treating each visit of Q to $(0,0)$ as a renewal event. From Eqs. (7.20) and (7.21), we have

$$\frac{\lim_{n \to \infty} \frac{1}{n} I(M, n)}{\lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}(Q[t] = 0)} = \frac{1}{\mathbb{E}(N_1)} = 1 - x. \tag{7.22}$$

Combing Eqs. (4.1), (7.17) and (7.22), and the fact that $\mathbb{E}(N_1) = \mathbb{E}(\text{Geo}(1 - x)) - 1 = \frac{1}{1-x} - 1$, we have

$$\frac{\lambda - (1-p)}{\lambda + 1 - p} \cdot \left[1 + x \cdot \frac{\lambda + 1 - p}{\lambda - (1-p)} \right] = 1 - x, \tag{7.23}$$

which yields

$$x = \frac{1-p}{\lambda + 1 - p}. \tag{7.24}$$

This completes the proof of Proposition 1. $\qquad \square$

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 $i \in \mathbb{Z}_+$,*

$$\mathbb{P}\left(Q[\infty] = i\right) = \left(1 - \frac{1-p}{\lambda}\right) \cdot \left(\frac{1-p}{\lambda}\right)^i. \qquad (7.25)$$

2. $\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} Q[i] = \mathbb{E}\left(Q\left[\infty\right]\right) = \frac{1-p}{\lambda - (1-p)}$, *a.s.*
3. *Let $E_i = \left\{m_i^{\Psi}, m_i^{\Psi} + 1, \ldots, m_{i+1}^{\Psi} - 1, m_{i+1}^{\Psi}\right\}$. Then the $|E_i|$ are i.i.d, with*

$$\mathbb{E}\left(|E_1|\right) = \frac{1}{\lim_{n \to \infty} \frac{1}{n} I\left(M^{\Psi}, n\right)} = \frac{\lambda + 1 - p}{\lambda - (1-p)}, \qquad (7.26)$$

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

$$\mathbb{P}\left(|E_1| \geq x\right) \leq a \cdot \exp\left(-b \cdot x\right). \qquad (7.27)$$

4.

$$m_i^{\Psi} \sim \frac{1}{\mathbb{E}\left(|E_1|\right)} \cdot i = \frac{\lambda - (1-p)}{\lambda + 1 - p} \cdot i, \quad a.s, \qquad (7.28)$$

   as $i \to \infty$.

PROOF. See Appendix A.4. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

7.3. *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 9, which we isolate here in the form of the following proposition.

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

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

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

1. Pver 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, i.e.,

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

and, analogously,

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

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 12.   **(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*

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

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

where $\Phi(Q, N) = \Phi(Q) \cap \{1, \ldots, N\}$ is the set of all deletable locations in Q in the first N slots, and $Q_{M^G}^k = D(Q^0, \{m_i^G : 1 \le i \le k\})$. 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 9. 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*

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

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

$$S\left(D\left(Q,M'\right),N\right) \geq \Delta_P\left(Q,N,m_1^G\right) + \min_{\substack{|\tilde{M}|=k-1, \\ \tilde{M} \subset \Phi\left(D(Q,m_1^G),N\right)}} S\left(D\left(Q_{M^G}^1,\tilde{M}\right),N\right).$$

$$(7.29)$$

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*

$$Q[n] \leq Q'[n], \quad \forall n \in \{1,\ldots,N\}.$$

Then, for any $K \geq 1$,

$$\min_{\substack{|M|=K, \\ M \subset \Phi(Q,N)}} S\left(D\left(Q,M\right),N\right) \leq \min_{\substack{|M|=K, \\ M \subset \Phi(Q',N)}} S\left(D\left(Q',M\right),N\right). \qquad (7.30)$$

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

$$\Delta\left(Q,N,M'\right) \geq \Delta\left(Q',N,M'\right). \qquad (7.31)$$

Recall the definition of a busy period in Eq. (7.2). Let $J(Q,N)$ be the total number of busy periods in $\{Q[n] : 1 \leq n \leq N\}$, with the additional convention $Q[N+1] \overset{\triangle}{=} 0$ so that the last busy period always ends on N. Let $B_j = \{l_j,\ldots,u_j\}$ 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 (c.f., Figure 3.2). 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

$$\Delta\left(Q,N,G(Q,N,1)\right) = \max_{1 \leq j \leq J(Q,N)} |B_j|. \qquad (7.32)$$

Let

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

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: $M' \cap \left(\cup_{j \in \mathcal{J}^*(Q,N)} B_j\right) \neq \varnothing$. If $l_{j^*} \in M'$ for some $j^* \in \mathcal{J}^*$, by Eq. (7.32), 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 Eq. (7.29) 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

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

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

$$\hat{Q}[n] \leq Q'[n], \quad \forall n \in \{1, \ldots, N\}. \tag{7.33}$$

Eq. (7.29) holds by combining Eq. (7.33) and Eq. (7.30) in Lemma 7, with $K = k - 1$.

Case 2: $M' \cap \left(\cup_{j \in \mathcal{J}^*(Q, N)} B_j \right) = \varnothing$. 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 Eq. (7.31) in Lemma 7, we have that[14]

$$\Delta\left(Q, N, M'\backslash\{m^*\}\right) \geq \Delta\left(D_P(Q, m^*), N, M'\backslash\{m^*\}\right). \tag{7.34}$$

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

$$\Delta_P\left(D\left(Q, M'\backslash\{m^*\}\right), N, m_1^G\right) = \max_{1 \leq j \leq J(Q, N)} |B_j| > \Delta_P(Q, N, m^*). \tag{7.35}$$

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

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

where (a) and (b) follow from Eqs. (7.34) and (7.35), respectively, which shows that Eq. (7.29) holds (and in this case the inequality there is strict).

Case 1 and 2 together complete the proof of Lemma 6.

$\square$

We are now ready to prove Proposition 3.

---

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

PROOF. (**Proposition 3**) Lemma 6 shows that, for any fixed number of deletions over a finite horizon N, the greedy deletion policy (Definition 12) yields the smallest area under the resulting sample path, Q, over $\{1, \dots, N\}$. 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*

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

Let $\{X_i : 1 \le i \le n\}$ 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 $H_n \lesssim \alpha n$ a.s. as $n \to \infty$ for some $\alpha \in (0, 1)$, we have

$$\limsup_{n \to \infty} f\left(\{X_i : 1 \le i \le n\}, H_n\right) \le \mathbb{E}\left(X_1 \cdot \mathbb{I}\left(X_1 \ge \overline{F}_{X_1}^{-1}(\alpha)\right)\right), \quad a.s., \quad (7.36)$$

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

PROOF. See Appendix A.5. □

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

$$l(n) = n - \max_{1 \le i \le I(M^{\Psi}, n)} |E_i| \tag{7.37}$$

where E_i is the ith deletion epoch of M^{Ψ}, defined in Eq. (7.3). Since $Q^0[n] \ge Q^0[m_i]$ for all $i \in \mathbb{N}$, it is easy to check that

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

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 $M^G = G\left(Q^0, n, I\left(M^{\Psi}, l(n)\right)\right)$. Then $m_i^G = m_i^{\Psi}$, for all $i \in \left\{1, \dots, I\left(M^{\Psi}, l(n)\right)\right\}$.*

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

$$
\begin{aligned}
&I\left(\tilde{M}, m_K^\Psi\right) \\
=\,&I\left(M^\Psi, l\left(m_K^\Psi\right)\right) + \left(I\left(M^\Psi, m_K^\Psi\right) - I\left(M^\Psi, l\left(m_K^\Psi\right)\right)\right) \\
&+ \left(I\left(\tilde{M}, m_K^\Psi\right) - I\left(M^\Psi, m_K^\Psi\right)\right) \\
=\,&I\left(M^\Psi, l\left(m_K^\Psi\right)\right) + \left(K - I\left(M^\Psi, l\left(m_K^\Psi\right)\right)\right) \\
&+ \left(I\left(\tilde{M}, m_K^\Psi\right) - I\left(M^\Psi, m_K^\Psi\right)\right) \\
=\,&I\left(M^\Psi, l\left(m_K^\Psi\right)\right) + h(K),
\end{aligned}
\tag{7.38}
$$

where

$$
h(K) = \left(K - I\left(M^\Psi, l\left(m_K^\Psi\right)\right)\right) + \left(I\left(\tilde{M}, m_K^\Psi\right) - I\left(M^\Psi, m_K^\Psi\right)\right).
\tag{7.39}
$$

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.6 $\square$

Let

$$
M^{G,n} = G\left(Q^0, n, I\left(\tilde{M}, n\right)\right),
\tag{7.40}
$$

where the greedy deletion map G was defined in Definition 12. By Lemma 9 and the definition of $M^{G,n}$, we have that

$$
M^\Psi \cap \left\{1, \ldots, l\left(m_K^\Psi\right)\right\} \subset M^{G, m_K^\Psi}.
\tag{7.41}
$$

Therefore, we can write

$$
M^{G, m_K^\Psi} = \left(M^\Psi \cap \left\{1, \ldots, l\left(m_K^\Psi\right)\right\}\right) \cup \overline{M}_K^G,
\tag{7.42}
$$

where $\overline{M}_K^G \triangleq M^{G, m_K^\Psi} \setminus \left(M^\Psi \cap \left\{1, \ldots, l\left(m_K^\Psi\right)\right\}\right)$. Since $\left|M^{G, m_K^\Psi}\right| = I\left(\tilde{M}, m_K^\Psi\right)$ by definition, by Eq. (7.38),

$$
\left|\overline{M}_K^G\right| = h(K).
\tag{7.43}
$$

We have

$$
\begin{aligned}
&S\left(D\left(Q^0, M^\Psi\right), m_K^\Psi\right) - S\left(D\left(Q^0, \tilde{M}\right), m_K^\Psi\right) \\
\overset{(a)}{\leq}\,&S\left(D\left(Q^0, M^\Psi\right), m_K^\Psi\right) - S\left(D\left(Q^0, M^{G, m_K^\Psi}\right), m_K^\Psi\right) \\
\overset{(b)}{=}\,&\Delta\left(D\left(Q^0, M^\Psi\right), m_K^\Psi, \overline{M}_K^G\right),
\end{aligned}
\tag{7.44}
$$

where (a) is based on the dominance of the greedy policy over any finite horizon (Lemma 6), and (b) follows from Eq. (7.42).

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

$$\limsup_{K \to \infty} \frac{\Delta\left(D\left(Q^0, M^\Psi\right), m_K^\Psi, \overline{M}_K^G\right)}{m_K^\Psi} \le g(\lambda), \quad a.s. \qquad (7.45)$$

Eqs. (7.44) and (7.45) combined imply that

$$
\begin{aligned}
C\left(p, \lambda, \pi_{NOB}\right) &= \limsup_{K \to \infty} \frac{S\left(D\left(Q^0, M^\Psi\right), m_K^\Psi\right)}{m_K^\Psi} \\
&\le g(\lambda) + \limsup_{K \to \infty} \frac{S\left(D\left(Q^0, \tilde{M}\right), m_K^\Psi\right)}{m_K^\Psi}, \\
&= g(\lambda) + \limsup_{n \to \infty} \frac{S\left(D\left(Q^0, \tilde{M}\right), n\right)}{n}, \quad a.s., \qquad (7.46)
\end{aligned}
$$

which shows that

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

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

To show Eq. (7.45), denote by Q the sample path after applying π_{NOB},

$$Q = D\left(Q^0, M^\Psi\right),$$

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 3.2. 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,

$$\mathbb{E}\left(V_1\right) \overset{(a)}{\le} \mathbb{E}\left(|E_1|^2\right) \overset{(b)}{<} \infty, \qquad (7.47)$$

where (a) follows from the fact that $|Q[n+1]-Q[n]| \le 1$ for all n, and hence $V_i \le |E_i|^2$ for any sample path of Q^0, and (b) from the exponential tail bound on $\mathbb{P}(|E_1| \ge x)$, given in Eq. (7.27).

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 (c.f., Figure 3.2). Therefore, the value of $\Delta\left(D\left(Q^0, M^\Psi\right), m_K^\Psi, \overline{M}_K^G\right)$ 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

$$
\begin{aligned}
&\limsup_{K\to\infty} \frac{\Delta\left(D\left(Q^0, M^\Psi\right), m_K^\Psi, \overline{M}_K^G\right)}{m_K^\Psi} \\
&= \limsup_{K\to\infty} f\left(\{V_i : 1 \le i \le K\}, h(K)\right) \cdot \frac{K}{m_K^\Psi} \\
&\overset{(a)}{=} \limsup_{K\to\infty} f\left(\{V_i : 1 \le i \le K\}, h(K)\right) \cdot \frac{\lambda+1-p}{\lambda-(1-q)} \\
&\overset{(b)}{=} \mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \ge \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda-(1-p)}\right)\right)\right) \cdot \frac{\lambda+1-p}{\lambda-(1-q)}
\end{aligned}
\tag{7.48}
$$

where (a) follows from Eq. (7.28), 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

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

as $\lambda \to 1$. Eq. (7.45) is proved by setting $g(\lambda) = \mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \ge \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda-(1-p)}\right)\right)\right) \cdot \frac{\lambda+1-p}{\lambda-(1-q)}$. This completes the proof of Proposition 3. $\qquad\square$

7.3.1. *Greedy v.s No-Job-Left-Behind Policies.* 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 don't we simply use the greedy rule as the offline policy of choice?

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 11). In contrast, the performance analysis of π_{NOB} in Section 7.2 reveals a highly *regenerative* structure: the deletions made by 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 11. 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 4.2.2), 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.

7.4. *Proof of Theorem 9.*

PROOF. **(Theorem 9)** Let $\{Q[n] : n \in \mathbb{Z}_+\}$be the resulting sample path after applying π_{NOB}. By Proposition 2, Q is a

The feasibility of π_{NOB} follows from Eq. (4.1), that, almost surely,

$$\lim_{n \to \infty} \frac{1}{n} I \left(M^\Psi, n \right) = \frac{\lambda - (1-p)}{\lambda + 1 - p} < \frac{p}{\lambda + 1 - p},$$

Let $\tilde{Q} = D \left(Q^0, M^\Psi \right)$ and $Q[n] = \tilde{Q} \left[n + m_1^\Psi \right]$, $\forall n \in \mathbb{N}$. Since $\lambda > 1 - p$, the random walk Q^0 is transient, and hence $m_1^\Psi < \infty$ almost surely. We have that

$$C \left(p, \lambda, \pi_{NOB} \right) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \tilde{Q}[i] = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{m_1^\Psi} \tilde{Q}[i] + \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n Q[i] = \frac{1-p}{p}, \tag{7.49}$$

almost surely. where the last equality follows from the second claim of Proposition 2, and the fact that $m_1 < \infty$ almost surely. Finally, the fact that $\lim_{\lambda \to 1} C \left(p, \lambda, \pi_{NOB} \right) = \lim_{\lambda \to 1} C^*_{\Pi_\infty} \left(p, \lambda \right)$ was proved in Proposition 3. This completes the proof of Theorem 9. $\qquad\square$

8. Policies with a Finite Lookahead.

8.1. *Proof of Theorem 11.*

PROOF. **(Theorem 11)** It suffices to focus on showing the feasibility of $\pi_{NOB}^{w(\lambda)}$. The performance guarantee of $\pi_{NOB}^{w(\lambda)}$ is an immediate consequence of Theorem 9 and the fact that the set of deletions made by $\pi_{NOB}^{w(\lambda)}$ is a superset of that made by π_{NOB}, as was pointed out in the discussion preceding Theorem 11. We begin by stating an exponential tail bound on the distribution of the discrete-time predictive window, $W(\lambda, n)$,

$$W(\lambda, n) = \max \left\{ k \in \mathbb{Z}_+ : T_{n+k} \leq T_n + w(\lambda) \right\}.$$

It is easy to see that $\left\{W\left(\lambda, m_i^{\Psi}\right) : i \in \mathbb{N}\right\}$ are i.i.d, with $W\left(\lambda, m_1^{\Psi}\right)$ distributed as Poisson with mean $(\lambda + 1 - p)w(\lambda)$. Since $\mathbb{P}\left(W\left(\lambda, m_1^{\Psi}\right) \geq x\right) \leq \mathbb{P}\left(\sum_{k=1}^{\lfloor w(\lambda) \rfloor} X_k\right)$, where the X_k's are i.i.d Poisson random variables with mean $\lambda + (1 - p)$, applying the Chernoff bound, we obtain that there exist $c, d > 0$ such that

$$\mathbb{P}\left(W\left(\lambda, m_1^{\Psi}\right) \geq \frac{\lambda + 1 - p}{2} \cdot w(\lambda)\right) \leq c \cdot \exp(-d \cdot w(\lambda)), \qquad (8.1)$$

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

For the purpose of analysis (as opposed to conceptual simplicity), we will consider a further relaxation from the policy $\pi_{NOB}^{w(\lambda)}$, denoted by $\sigma^{w(\lambda)}$.

DEFINITION 13. *Fix* $w \in \mathbb{R}_+$. σ^w *is a deletion policy, such that for each deletion epoch* E_i, $i \in \mathbb{N}$,

1. *if* $|E_i| \leq W\left(\lambda, m_i^{\Psi}\right)$, *then only the arrival in slot* m_i^{Ψ} *is deleted;*
2. *otherwise, all arrivals within the epoch* E_i *are deleted.*

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

Denote by D_i the number of deletions made by $\sigma^{w(\lambda)}$ in the ith epoch. By the construction of the policy, the D_i's are i.i.d, and depend only on the length of E_i and the number of arrivals within. We have[15]

$$
\begin{aligned}
&\mathbb{E}\left(D_1\right) \\
&\leq 1 + \mathbb{E}\left[|E_i| \cdot \mathbb{I}\left(|E_i| \geq W\left(\lambda, m_i^{\Psi}\right)\right)\right] \\
&\leq 1 + \mathbb{E}\left[|E_i| \cdot \mathbb{I}\left(|E_i| \geq \frac{\lambda + 1 - p}{2} \cdot w(\lambda)\right)\right] \\
&\quad + \mathbb{E}\left(|E_i|\right) \cdot \mathbb{P}\left(W\left(\lambda, m_i^{\Psi}\right) \leq \frac{\lambda + 1 - p}{2} \cdot w(\lambda)\right) \\
&\leq 1 + \left(\sum_{k=\frac{\lambda+1-p}{2} \cdot w(\lambda)}^{\infty} k \cdot a \cdot \exp(-b \cdot k)\right) + \frac{\lambda}{\lambda - (1 - p)} \cdot c \cdot \exp(-d \cdot w(\lambda)) \\
&\overset{(a)}{\leq} 1 + h \cdot w(\lambda) \cdot \exp(-l \cdot w(\lambda)), \qquad (8.2)
\end{aligned}
$$

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

---

[15]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 $w(\lambda)$.

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

$$\frac{\mathbb{E}(D_1)}{\mathbb{E}(E_1)} = \frac{\mathbb{E}(D_1)}{\lambda} \leq \frac{p}{\lambda + 1 - p}. \tag{8.3}$$

By Eqs. (8.2) and (8.3), we have that

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

which yields, after taking the logarithm on both sides,

$$w(\lambda) \geq \frac{1}{b} \log\left(\frac{1}{1-\lambda}\right) + \frac{1}{b} \log\left(\frac{[\lambda - (1-p)] \cdot h \cdot w(\lambda)}{1-p}\right). \tag{8.4}$$

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 $\pi_{NOB}^{w(\lambda)}$ is also feasible. This completes the proof of Theorem 11. $\qquad\square$

9. Conclusion 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 also serves as a simple canonical model for studying resource pooling in large server farms and large clusters [17] (Section 5). 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 11).

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 heavy-traffic delay collapse can be proved (Conjecture 1), which, together with the upper bound given in Theorem 11, 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, since our analysis of π_{NOB} depends primarily on *macroscopic* properties of the initial sample path Q^0 (e.g., lengths of deletion epochs), and it may be possible to establish the generalizations 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 of 1 and by characterizing the resulting queue length process of π_{NOB} as a reflected Brownian motion in $\mathbb{R}_+$, in the limit of $\lambda \to 1$ and $p \to 0$, with appropriate scaling.

There are other issues that need to be better understood if the methodology presented here is to be used 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 noise or perturbation (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 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 3.4).

REFERENCES

[1] E. Altman and A. Swartz, "Markov decision problems and state-action frequencies," *SIAM J. Control and Optimization*, 29(4):786-809, July 1991.

[2] F. J. Beutler and K. W. Ross, "Time-average optimal constrained semi-Markov decision processes," *Adv. Appl. Prob.*, 18:341-359, 1986.

[3] R. G. Gallager, "Discrete Stochastic Processes", *Kluwer*, Boston, 1996.

[4] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and non-parametric models for traffic flow forecasting", *Cold Spring Harbor Symp. Quant. Biol.* , 10 (4), 303?321.

[5] S. Stidham Jr., "Optimal control of admission to a queueing system," *IEEE Trans. Auto. Control,* (AC-30):705-713, 1985.

[6] N. W. Nawijn, "Look-Ahead policies for admission to a single server loss system," *operations research*, 38:854-862, 1990.

[7] M. Carr and B. Hajek, "Scheduling with asynchronous service opportunities with applications to multiple satellite systems," *IEEE Trans. Automatic Control*, 38:1820 - 1833, 1993.

[8] E. G. Coffman, Jr., P. Jelenkovic and B. Poonen, "Reservation probabilities," *Adv. Perf. Anal.* 2:129-158, 1999.

[9] Y. Lu and A. Radovanovic, "Asymptotic blocking probabilities in loss networks with subexponential demands," *J. Appl. Probab.*, 44(4):1088-1102, 2007.

[10] R. Levi and C. Shi, "Revenue management of reusable resources with advanced reservations," submitted, 2011.

[11] S. C. Kim and I. Horowitz, "Scheduling hospital services: The efficacy of elective surgery quotas," *Omega*, 30:335-346, 2002.

[12] M. Fisher and A. Raman, "Reducing the cost of demand uncertainty through accurate response to early sales," *Operations Research*, 44(1):87-99, 1996.

[13] A. Mandelbaum and M. I. Reiman, "On pooling in queueing networks," *Management Science*, 44(7):971-981, 1998.

[14] J. M. Harrison and M. J. Lopez, "Heavy traffic resource pooling in parallel-server systems," *Queueing Systems*, 33:39-368, 1999.

[15] S. L. Bell and R. J. Williams, "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(3): 608-649, 2001.

[16] A. Mandelbaum and A. L. Stolyar, "Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule," *Operations Research*, 52(6):836-855, 2004.

[17] J. N. Tsitsiklis and K. Xu, "On the power of (even a little) resource pooling," Stochastic Systems, 2:1-66 (electronic), 2012.

APPENDIX A: ADDITIONAL PROOFS

A.1. Proof of Lemma 2.

PROOF. **(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 prove 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

$$\tilde{Q}[n] = \tilde{Q}[N] + (A[n] - A[N]) - (S[n] - S[N]) - I\left(M^{\Psi}, n\right), \qquad (A.1)$$

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 Eq. (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., $\tilde{Q}[n] \geq 0$ for all n, and $\tilde{Q}[N] < \infty$ a.s. $\qquad \square$

A.2. Proof of Lemma 4.

PROOF. **(Lemma 4)**

1. Recall the point-wise deletion map, $D_P(Q, n)$, defined in Definition 2. For any 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

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

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

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

Note that Eq. (A.3) is implied by (and in fact, equivalent to) the definition of the m_k's (Definition 8), 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 $\mathbb{P}\left(Q^0[t] \neq Q^0[t-1] \mid Q^0[t-1] > 0\right) = 1$ for all $t \in \mathbb{N}$ (c.f., Eq. (2.1)), at least one deletion occurred in $\{n - 1 + m_1, n + m_1\}$. If the deletion occurred on $n + m_1$, we are done. Suppose a deletion occurred 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 have also occurred 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

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

Since the random walk Q^0 is transient and whose step sizes are 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

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

where (a) follows from Eq. (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 on m_i in Q^0, and hence $Q^0[n-1+m_1] = Q^0[n+m_1] - 1$. Therefore, by the definition of m_i,

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

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

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

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

where (a) follows from the second claim (c.f., Eq. (A.5)), (b) from the fact that there is no deletion over $\{I(M, n+m_1), \ldots, n+m_1\}$, and (c) from the fact that $n + m_1 \geq I(M, n+m_1)$ and Eq. (3.3).

$\square$

A.3. Proof of Lemma 5.

PROOF. (**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

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

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

$$\mathbb{P}\left(X[n+1] = x_1 + 1 \,\middle|\, X[n] = x_1, \min_{r \geq n+1} X[r] = 0\right)$$

$$= \frac{\mathbb{P}\left(X[n+1] = x_1 + 1, \min_{r \geq n+1} X[r] = 0 \,\middle|\, X[n] = x_1\right)}{\mathbb{P}\left(\min_{r \geq n+1} X[r] = 0 \,\middle|\, X[n] = x_1\right)}$$

$$\stackrel{(a)}{=} \frac{\mathbb{P}\left(X[n+1] = x_1 + 1 \,\middle|\, X[n] = x_1\right) \cdot \mathbb{P}\left(\min_{r \geq n+1} X[r] = 0 \,\middle|\, X[n+1] = x_1 + 1\right)}{\mathbb{P}\left(\min_{r \geq n+1} X[r] = 0 \,\middle|\, X[n] = x_1\right)}$$

$$\stackrel{(b)}{=} q \cdot \frac{h(x_1 + 1)}{h(x_1)}, \tag{A.7}$$

where

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

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 Eqs. (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 Eq. (A.7), this implies that

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

which proves the claim. $\qquad\square$

A.4. Proof of Proposition 2.

PROOF. (**Proposition 2**) 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 distribution of 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 $\mathbb{E}(Q[\infty])$ a.s.

The fact that E_i's are i.i.d was shown in the discussion preceding Eq. (7.20) in the proof of Proposition 1. The value of $\mathbb{E}(|E_1|)$ follows by combining Eqs. (4.1) and (7.20).

Let $B_{i,j}$ be the length of the jth busy period (defined in Eq. (7.2)) 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,

$$\lim_{\theta \downarrow 0} G_{B_{1,1}}(\theta) = 1, \tag{A.10}$$

By Eq. (7.19), we have

$$
\begin{aligned}
G_{|E_1|}(\epsilon) &= \mathbb{E}\left(\exp\left(\epsilon \cdot |E_1|\right)\right) \\
&= \mathbb{E}\left(\exp\left(\epsilon \cdot \left(1 + \sum_{j=1}^{N_1} B_{1,j}\right)\right)\right) \\
&\overset{(a)}{=} \mathbb{E}\left(e^{\epsilon}\right) \cdot \mathbb{E}\left(\exp\left(N_1 \cdot G_{B_{1,1}}(\epsilon)\right)\right) \\
&= \mathbb{E}\left(e^{\epsilon}\right) \cdot G_{N_1}\left(\ln\left(G_{B_{1,1}}(\epsilon)\right)\right), \tag{A.11}
\end{aligned}
$$

where (a) follows from the fact that $\{N_1\} \cup \{B_{1,j} : j \in \mathbb{N}\}$ are mutually independent, and $G_{N_1}(x) = \mathbb{E}\left(\exp\left(x \cdot N_1\right)\right)$. Since $N_1 \overset{d}{=} \mathrm{Geo}(1 - x) - 1$, $\lim_{x \downarrow 0} G_{N_1}(x) = 1$, and by Eq. (A.10), we have that $\lim_{\epsilon \downarrow 0} G_{|E_1|}(\epsilon) = 1$, which implies Eq. (7.27).

Finally, Eq. (7.28) follows from the third claim and the Elementary Renewal Theorem. □

A.5. Proof of Lemma 8.

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

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\right) = \mathbb{E}\left(\mathbb{I}\left(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\right)\right) < \alpha, \quad a.s. \tag{A.12}$$

Denote by $S_{n,k}$ set of top k elements in $\{X_i : 1 \leq i \leq n\}$. By Eq. (A.12) and the fact that $H_n \lesssim \alpha n$ a.s., there exists $N > 0$ such that

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

which implies that

$$
\begin{aligned}
&\limsup_{n \to \infty} f\left(\{X_i : 1 \leq i \leq n\}, H_n\right) \\
&\leq \limsup_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} X_i \cdot \mathbb{I}\left(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\right) \\
&= \mathbb{E}\left(X_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{X_1}^{-1}(\alpha)\right)\right) \quad a.s., \tag{A.13}
\end{aligned}
$$

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

A.6. Proof of Lemma 10.

PROOF. (**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*

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

as $n \to \infty$.

PROOF.

$$\begin{aligned}
\lim_{n \to \infty} \mathbb{P}\left(\max_{1 \leq i \leq n} X_i \leq \frac{2}{b} \ln n\right) &= \lim_{n \to \infty} \mathbb{P}\left(X_1 \leq \frac{2}{b} \ln n\right)^n \\
&\leq \lim_{n \to \infty} \left(1 - a \cdot \exp(-2 \ln n)\right)^n \\
&= \lim_{n \to \infty} \left(1 - \frac{a}{n^2}\right)^n \\
&= 1. \quad\quad\quad\quad\quad\quad (A.14)
\end{aligned}$$

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

$$m_K^\Psi = \sum_{i=0}^{K-1} |E_i| \sim \mathbb{E}(|E_1|) \cdot K = \frac{\lambda + 1 - p}{\lambda - (1-p)} \cdot K, \quad a.s., \quad\quad (A.15)$$

by the strong law of large numbers. By Lemma 11 and Eqs. (7.27), we have

$$\max_{1 \leq i \leq K} |E_i| = o(K), \quad a.s., \quad\quad\quad\quad (A.16)$$

as $K \to \infty$. By Eq. (A.16) and the fact that $I\left(M^\Psi, m_K^\Psi\right) = K$, we have

$$\begin{aligned}
K - I\left(M^\Psi, l\left(m_K^\Psi\right)\right) &= K - I\left(M^\Psi, m_K^\Psi - \max_{1 \leq i \leq K} |E_i|\right) \\
&\overset{(a)}{\leq} K - I\left(M^\Psi, m_K^\Psi\right) + \max_{1 \leq i \leq K} |E_i| \\
&= \max_{1 \leq i \leq K} |E_i| \\
&= o(K), \quad a.s., \quad\quad\quad\quad (A.17)
\end{aligned}$$

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 $\tilde{M}$ is feasible,

$$I\left(\tilde{M}, n\right) \lesssim \frac{p}{\lambda + 1 - p} \cdot n, \tag{A.18}$$

as $n \to \infty$. We have,

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

as $K \to \infty$, where (a) follows from Eqs. (A.15) and (A.18), (b) from Eqs. (A.15) and (A.17), which completes the proof. $\qquad\square$

E-mail: spencer@courant.nyu.edu   E-mail: madhu@mit.edu

E-mail: kuangxu@mit.edu