Processor-Ring Communication:
A Tight Asymptotic Bound on Packet Waiting Times

E. G. Coffman, Jr.,[1] *Nabil Kahale*[2] and *F. T. Leighton*[3]

[1]Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974
[2]AT&T Research Laboratories, Murray Hill, NJ 07974
[3]Dept. of Math. and Laboratory for Computer Science, MIT, Cambridge, MA 02139

July 18, 1996

ABSTRACT

We consider N processors communicating unidirectionally over a closed transmission channel, or ring. Each message is assembled into a fixed-length packet. Packets to be sent are generated at random times by the processors, and the transit times spent by packets on the ring are also random. Packets being forwarded, i.e., packets already on the ring, have priority over waiting packets. The objective of this paper is to analyze packet waiting times under a greedy policy, within a discrete Markov model that retains the over-all structure of a practical system, but is simple enough so that explicit results can be proved. Independent, identical Bernoulli processes model message generation at the processors, and i.i.d. geometric random variables model the transit times. Our emphasis is on asymptotic behavior for large ring sizes, N, when the respective rate parameters have the scaling λ/N and μ/N. Our main result shows that, if the traffic intensity is fixed at $\rho = \lambda/\mu < 1$, then as $N \to \infty$ the expected time a message waits to be put on the ring is bounded by a constant. This result verifies that the expected waiting time under the greedy policy is within a constant factor of that under an optimal policy.

Processor-Ring Communication: A Tight Asymptotic Bound on Packet Waiting Times

E. G. Coffman, Jr.,[1] *Nabil Kahale*[2] and *F. T. Leighton*[3]

[1]AT&T Bell Laboratories, Murray Hill, NJ 07974
[2]DIMACS, Rutgers University, New Brunswick, NJ 08855
[3]Dept. of Math. and Laboratory for Computer Science, MIT, Cambridge, MA 02139

1. Introduction

Communication among N processors takes place counterclockwise along a slotted circular transmission channel, or *ring.* A processor generates messages, receives messages, and forwards messages between other processors. Each message is a *packet* of fixed duration. One time unit is required for a packet to be sent or forwarded from one processor to its counterclockwise neighbor. Packets are generated randomly at the processors according to i.i.d. arrival processes. The integer times spent by packets on the ring, packet *transit times*, are i.i.d. random variables. Packets being forwarded on the ring have priority; while a processor has a packet to be forwarded, it can not place one of its own waiting packets on the ring. A packet waiting for transmission is held in a queue at the processor where it was generated.

The details defining a practical implementation of a processor ring are many and varied. Indeed, the applications and analysis of communication rings form a rather large and growing literature; see van Arem and van Doorn (1990), Barroso and Dubois (1993), and Georgiadis, Szpankowski, and Tassiulas (1993) for brief surveys and many references. As a concession to mathematical tractability, we adopt here the simple discrete Markov model in Fig. 1, where the ring is partitioned into *cells*, each capable of holding a single packet. The cells rotate counterclockwise past the processors in discrete steps, one step per unit of time. Packets are generated at each of the N processors by a Bernoulli process at rate λ/N $0 < \lambda < N$, per time unit (step); the total arrival rate is then λ. The packet transit times are geometrically distributed with rate parameter μ/N, $N > \mu > \lambda$. Thus, at any given step, a packet on the ring departs with probability μ/N and stays for at least one more step with probability $1-\mu/N$, independent of how long the packet has already been on the ring.

We will explain shortly the reason for the scaling of arrival and transit-time parameters by the ring size.

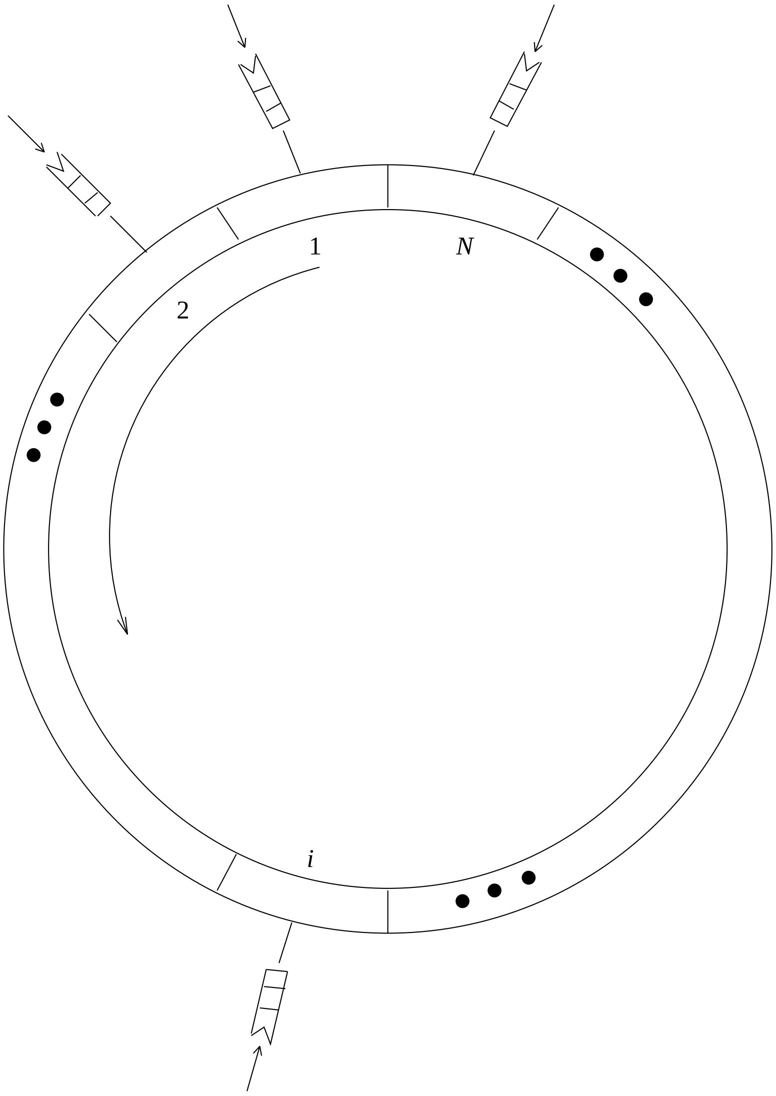

Figure 1: The rotating ring model.

In each step, the ring system undergoes a transition according to the following sequence:

(i) The ring rotates one position while processor queues accept new arrivals, if any (at most one per queue in each step).

(ii) Packets on the ring that have completed their transit times are delivered, i.e., removed from their cells.

(iii) Each processor with a nonempty queue opposite an empty cell then puts a waiting packet into this cell.

This gives the *nonblocking* model; reversing (ii) and (iii) would give the *blocking* model: a departing packet can not be replaced in the same time step by a waiting packet. As we shall see, our asymptotic results apply to both models. The above sequence gives the *greedy* cell admission policy, placing waiting packets on the ring as soon as empty cells are available.

As discussed in Coffman et al. (1993), the greedy policy has the undesirable effect of occasionally "freezing out" certain processor queues for long periods of time; long trains of occupied cells pass by such processors denying them access to the ring. The results of this paper will show that, for large rings within our probability model, the greedy rule is remarkably efficient, and that in fact the above behavior is quite rare.

Our specific objective is to analyze packet waiting times under the greedy policy. (Hereafter, unless noted otherwise, waiting times always refer to times spent waiting in processor queues.) To prepare for the statement of our main theorem on waiting times, we need a little more notation. For a given admission policy, we denote the joint queue length at integer time t by $\mathbf{Q}(t) = (Q_1(t), \ldots, Q_N(t))$, where $Q_i(t)$ is the number in the i^{th} processor queue at time t. The phrase 'at time t' means at an instant just after t so that events, if any, occurring at t have already taken place. Define the N-bit vector $\mathbf{R}(t)$ whose i^{th} bit is 1 if and only if a packet is in the i^{th} cell at time t. Hereafter, the term *state* refers to a pair $\mathbf{Q}(t), \mathbf{R}(t)$ at some time t. It follows from the geometric law for transit times that the *ring process* $\{(\mathbf{Q}(t), \mathbf{R}(t)), t = 0, 1, \ldots\}$ is a Markov chain. It was shown by Coffman et al (1993) that, if $\lambda < \mu$, then the ring process under the greedy rule is ergodic. Unfortunately, an exact analysis of the stationary behavior of this ring process seems quite difficult. Indeed, attempts to solve the balance equations have so far failed even for the case $N = 2$. Thus, we turn to asymptotic estimates for large ring sizes, N, with λ and μ fixed and $\lambda < \mu$. That is why we introduced the scalings λ/N and μ/N; as we allow N to increase, the traffic intensity will remain fixed at $\rho = \lambda/\mu$, the usual product of arrival rate and average service (transit) time.

With $\lambda < \mu$, let Q have the stationary distribution common to all queue lengths $Q_i(t)$, and let W be the waiting time of a packet in the stationary regime.

Theorem 1.1. *Fix λ and μ with $\lambda < \mu$. Then under the greedy policy*

$$E[Q] = \Theta(1/N) \ .$$

Thus, by Little's theorem,

$$E[W] = \Theta(1) \ .$$

The lower bounds are easy to see, as follows. Consider the entire ring as an N-server system with a total arrival rate λ and maximum departure rate μ. Then by Little's theorem, the arrival rate λ times the average time spent on the ring, i.e., N/μ, must be equal to the expected number of packets on the ring in the stationary regime, i.e., ρN. But if a positive fraction $\rho > 0$ of the ring is occupied on average, then there must be a positive average waiting time $E[W] = \Omega(1)$ to get on the ring and hence $E[Q] = \Omega(1/N)$.

In the usual way, *on-line* admission policies are those deciding packet admissions solely on the basis of information currently available about packets already in the system, waiting or on the ring. Such information can include, for example, queue lengths and the elapsed times already spent in the system by packets. As we will see later, it is convenient to extend this class of admission policies by allowing decisions to depend also on the times and queues of future arrivals. *Hereafter, unless stated otherwise, the term* policy *refers to a policy in this extended class.* Note in particular that policies retain the on-line property with respect to transit times on the ring; i.e., we do not allow policies that base decisions on prior knowledge of remaining transit times.

We say that a policy A is optimal if, over any interval $[0, T]$, the sum of waiting times (in queue) under A is stochastically no larger than that under any other policy starting in the same initial state. We prove in the next section that the greedy policy is an optimal policy (the proof will need the geometric law for transit times). In the proof of Theorem 1.1, this result allows us to analyze a more tractable policy with the same asymptotic performance as the greedy rule; the more tractable policy exploits the fact that policies can base decisions on the times and queues of future arrivals.

Coffman et al. (1993) proved in an earlier paper that the growth of the expected waiting time in our model was sublinear in N, i.e., $E[W] = o(N)$. Our much stronger result shows that the expected waiting time is in fact bounded by a constant. So by Little's theorem, an important practical implication of our result is that the expected size of a buffer needed to hold all waiting packets is bounded by a constant uniformly in N. The proof of Theorem 1.1 requires a much more intricate probabilistic analysis than the one in Coffman et al. (1993), where the law of large numbers was the basic tool. Here, we will need more powerful asymptotic bounds (e.g., those of Chernoff type) on the tail probabilities for sums of independent random variables and the excursions of Lindley processes (see e.g. Prabhu (1965), p. 66); these appear as lemmas in Section 3. The proof of the upper bound $E[Q] = O(1/N)$ is given in Sections 4 and 5. The paper concludes in Section 6 with a brief discussion of extensions and open problems.

2. Preliminaries

Consider the packet at the head of any given nonempty queue. Since transit times are geometrically distributed with parameter μ/N, the probability that this packet is placed on the ring in the current time step is at least μ/N; the conditional probability is precisely μ/N if the cell is occupied on arrival and it is trivially 1 if the cell is empty. Thus, one expects that, in statistical equilibrium, the i^{th} queue length Q_i is bounded stochastically for each i by the length of a single-server Markov (i.e., M/M/1) queue in discrete time with arrival and service rate parameters λ/N and μ/N. Moreover, this bound should hold independently for each queue. Indeed, these observations are but a special case of Theorem 2 in Coffman et al. (1993). An easy analysis of the discrete-time M/M/1 queue then proves

Lemma 2.1. *For each i independently, Q_i is stochastically smaller than a non-negative integer random variable L with $P(L = n) \sim (1-\rho)\rho^n$ as $N \to \infty$ for every $n \geq 0$, and with*

$$P(L > n) = O(e^{-\nu n}) \ , \tag{2.1}$$

where $\nu = \ln 1/\rho > 0$.

Hereafter, we take the equivalent point of view that *the queues rotate past the ring of cells, which remains fixed.* As shown in Fig. 2, in any given time interval $[0, T]$, the ring process can be represented by events on a cylindrical lattice cut at some cell position and laid out as a rectangle. For simplicity, we assume that the cylinder is cut between cell N

and cell 1. Along the top of the rectangle the $Q_i(0)$, $1 \leq i \leq N$, give the initial state of the queues, and the bullets ($\bullet$'s) indicate the initial cell states: a cell with a $\bullet$ at time 0 is empty, otherwise, it is occupied. Again for simplicity, we assume queue 1 is at cell 1 at time 0. Within the rectangle, circles ($\circ$'s) and bullets give a random sample of arrivals and departures, respectively. A $\bullet$ and $\circ$ can appear at the same lattice point; the probability of such an event is $O(1/N^2)$ and hence relatively low; for simplicity, the figures in this paper do not show samples where such coincidences occur.

The greedy policy is represented by a suitable assignment of cells to circles (new arrivals) and to packets in the initial state. An example is shown in Fig. 2. The *motion* lines drawn between packets and assigned cells describe the trajectories of the packets in time and space; their vertical components correspond to waiting times. A motion line is broken into two pieces when it extends past cell N, one ending at the right boundary, and one beginning at the same time at the left boundary.

To ensure that an assignment of circles and initial packets to cells is valid, one must check to see that the cell, say c, at which a motion line terminates, at time t say, is indeed empty at time t. Thus, if t', $0 \leq t' < t$, is the time of the last departure (bullet) in cell c, then no other motion lines can terminate at cell c in the interval $[t', t]$.

We conclude this section with a proof that greedy is optimal in that it minimizes stochastically the sum S of waiting times over any given interval $[0, T]$. The proof uses the following simple relation between S and the queue lengths $Q_i(t)$ during $[0, T]$:

$$S = \sum_{t=0}^{T} \sum_{i=1}^{N} Q_i(t) \ . \tag{2.2}$$

Theorem 2.1. *The greedy rule is an optimal admission policy.*

Proof: Consider ring operation over an interval $[0, T]$, and let A be an arbitrary policy. To compare total waiting times in $[0, T]$ under A and greedy, both starting in the same initial state, we compare both to an intermediate algorithm A*, which is artificial in that it sometimes returns packets to queues before they have completed their transit times. Under A* an occupied queue places a packet into the first available empty cell, just as with the greedy policy. But suppose that, under A* at some time t, a cell c occupied by packet φ is

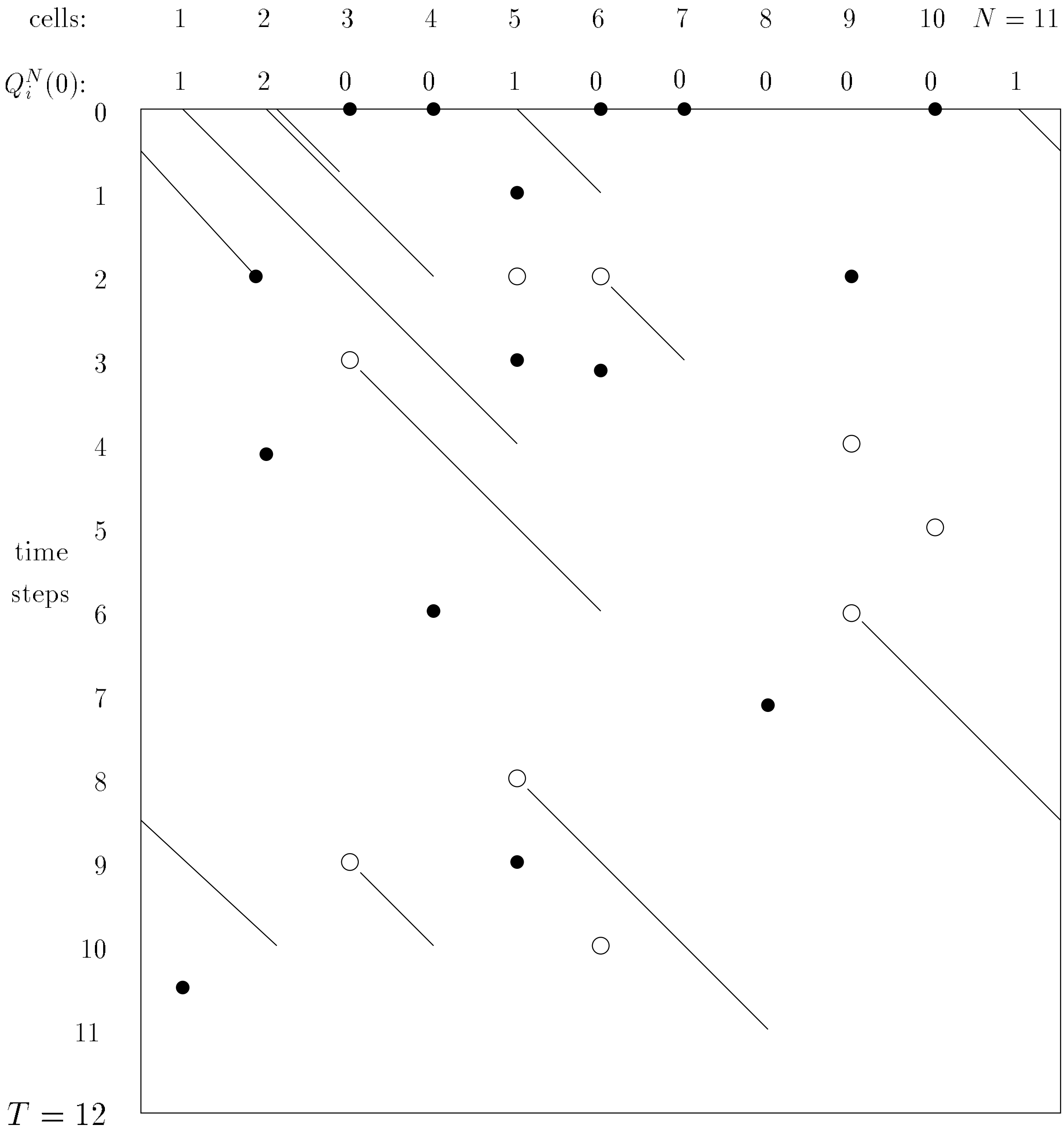

Figure 2: Greedy Assignment

in front of a queue having a packet φ' that, under A, would have been in c at time t. Then at time t, φ and φ' change places under A*; φ joins the queue and φ' enters the cell.

At any given queue, the admissions under A* and greedy implement the same deterministic rule except at times when A* exchanges a packet in the queue with the packet in the cell in front of the queue. *But such an exchange does not change the state (any queue length or the state of any cell) of the ring process*; from the point of view of the ring process, the exchange has the effect of doing nothing, which is just what the greedy rule would do in the same circumstances. Thus, if A* and greedy start in the same initial state, then the joint queue-length process over $[0,T]$, and hence by (2.2) the sum of waiting times over $[0,T]$, is stochastically the same under A* and greedy.

It remains to show that the sum of waiting times over $[0,T]$ under A is at least as large stochastically as it is under A*. In fact, we prove the stronger deterministic result: For a given initial state, a given sequence of arrivals over $[0,T]$, and a given sample of the remaining transit times of all packets in the system during $[0,T]$, the sum of waiting times under A is at least that under A*. To see this, note first that, although A* may put a packet φ on and off the ring several times, eventually one of three events will occur: φ will depart, T steps will have been taken, or φ will be in a queue when the cell assigned to it under A catches up to it. In the last case, A* places φ on the ring making an exchange, if needed, and leaves it there until it departs or T steps have been made. Thus, every packet during $[0,T]$ has moved along the ring under A* at least as far as it has moved under A, and so the sum of waiting times under A* is deterministically at most the sum under A. ■

3. Probability Bounds[†]

We begin with a useful Chernoff bound that combines Theorems A.12 and A.13, pp. 237–238, in Alon and Spencer (1991).

Lemma 3.1. *Let $Z = Z_1 + \ldots + Z_n$, where the Z_i are independent Bernoulli random variables with $P(Z_i = 1) = p_i$, $P(Z_i = 0) = 1 - p_i$. Then for any $\epsilon > 0$, there exists a $\beta > 0$*

[†] The reader may wish to skip this section at first reading, referring back to it as needed while reading Section 5.

such that

$$P((1-\epsilon)E[Z] < Z < (1+\epsilon)E[Z]) = 1 - O(e^{-\beta E[Z]}) \ . \tag{3.1}$$

Next, we consider a Lindley process, starting at the origin and defined by (x^+ denotes the positive part of x)

$$\zeta_0 = 0, \quad \zeta_i = (\zeta_{i-1} + U_i)^+ \ , \tag{3.2}$$

with $U_i = X_i - Y_i$, where $\{X_i\}$ and $\{Y_i\}$ are independent sequences of i.i.d. random variables. In our application, Y_i is an integer in $\{0, \ldots, K\}$ with K a given integer constant independent of N, and X_i is the number of arrivals of a rate-$a\lambda/N$ Bernoulli process in bN time steps, where a and b are constants independent of N. Thus, for large N, X_i is approximately Poisson distributed with mean $ab\lambda$. It is easy to check that X_i and hence U_i has an exponential tail probability, i.e., there exists a $\kappa > 0$ such that

$$P(U_i > x) \leq P(X_i > x) = O(e^{-\kappa x}) \ . \tag{3.3}$$

The process $\{\zeta_i\}$ is said to have negative drift if $E[Y_i] > E[X_i]$ and hence $E[U_i] < 0$. The next result follows from standard theory (e.g., see Asmussen (1987)). Let the X_i and U_i be distributed as X and U, respectively.

Lemma 3.2. *If $E[U] < 0$, then $E[\zeta_i]$ is bounded by a constant uniformly in $i \geq 0$. The distributions of the ζ_i converge in total variation to the distribution of a random variable ζ with moments of all orders.*

In addition to Lemma 3.2, we will need certain probability bounds on excursions of $\{\zeta_i\}$. These will be derived in terms of corresponding bounds for the unrestricted process

$$\xi_i = \xi_{i-1} + U_i, \quad i \geq 1 \ , \tag{3.4}$$

with the U_i defined as before, and with a given initial state ξ_0. Hereafter, we assume a negative drift $E[U] < 0$.

The probability bound on excursions of $\{\xi_i\}$ that we will use in the analysis of $\{\zeta_i\}$ is developed as follows. Since $E[U] < 0$, and $P(U > 0) > 0$, there exists an $\alpha_0 > 0$ such that $E[e^{\alpha_0 U}] = 1$. Define the process $\xi_i^* = e^{\alpha_0 \xi_i}$, $i \geq 0$, with the property

$$E[\xi_{i+1}^* \mid \xi_i^*] = E[e^{\alpha_o U_i} \xi_i^* \mid \xi_i^*] = \xi_i^* E[e^{\alpha_0 U}] = \xi_i^* \ .$$

Together with our assumptions on U, this shows that $\{\xi_i^*\}$ is a uniformly integrable martingale, so we have

$$\begin{aligned} P\left(\sup_{i\geq 0} \xi_i \geq x\right) &= P\left(\sup_{i\geq 0} \xi_i^* \geq e^{\alpha_0 x}\right) \\ &\leq e^{-\alpha_0 x} E[\xi_0^*] = E[e^{-\alpha_0(x-\xi_0)}] \ , \end{aligned} \tag{3.5}$$

where the inequality follows from Doob's martingale inequality (see, for example, Section 35 in Billingsley (1986)).

We now use (3.5) to get similar bounds for the *busy periods* of $\{\zeta_i\}$. In analogy with queueing applications, we say that steps i_1 through i_2, $i_2 > i_1$, comprise a busy period if $\{\zeta_i\}$ moves away from the origin at step $i_1 \geq 1$ and makes its first subsequent return to the origin at step i_2, i.e., $\zeta_{i_1 - 1} = 0$, $\zeta_j > 0$, $i_1 \leq j < i_2$, and $\zeta_{i_2} = 0$. The process is *idle* while it resides at the origin. We want a probability bound on the maximum value of the process during a busy period B. For this purpose, we make use of the fact that, away from the origin, $\{\zeta_i\}$ behaves as an unrestricted random walk. In particular, the conditional probability that, given the first jump $U_{i_1} > 0$, $\{\zeta_i\}$ exceeds level x before its next return to the origin is the same as the probability that, starting in state U_{i_1}, the unrestricted version $\{\xi_i\}$ exceeds level x before its first passage to a point at or below the origin. As an easy consequence of (3.3) and (3.5), we have that, for a randomly chosen busy period B of $\{\zeta_i\}$,

$$P\left(\sup_{i\in B} \zeta_i > x\right) \leq E[e^{-\alpha_0(x-U)} | U > 0] \leq e^{-\alpha_0 x} E[e^{\alpha_0 X} | X > K] = O(e^{-\alpha_0 x}), \tag{3.6}$$

since $X \geq U \geq X - K$ and X is a binomial random variable with a mean bounded independently of N.

Our primary interest is in the behavior of $\{\zeta_i\}$ over a finite (and large) number of steps. It is convenient to let N denote the number of steps, since in later applications of the results below, N will also denote the ring size. For example, a bound on $P(\sup_{1\leq i\leq N} \zeta_i > \alpha \ln N)$, $\alpha > 0$, will be useful. To get such a bound, note that there are at most $N/2$ busy periods in the first N steps of $\{\zeta_i\}$. Then by (3.6)

$$P\left(\sup_{1\leq i\leq N} \zeta_i > x\right) \leq \frac{N}{2} P\left(\sup_{i\in B} \zeta_i > x\right) = O(e^{-\alpha_0 x + \ln N}) \ .$$

Thus, for any $\gamma > 0$, we can choose $x = x(N) = \alpha \ln N$ with $\alpha = \alpha(\gamma)$ sufficiently large that

$$P\left(\sup_{1 \le i \le N} \zeta_i > \alpha \ln N\right) = O(e^{-\gamma \ln N}) = O(N^{-\gamma}) \ . \tag{3.7}$$

Consider next the duration D of busy period B.

Lemma 3.3. *There exists an* $\eta_0 > 0$ *such that*

$$P(D > y) = O(e^{-\eta_0 y}) \ .$$

Proof: Let $\{U_i\}$ be the common sequence generating both $\{\zeta_i\}$ and $\{\xi_i\}$, $\zeta_0 = \xi_0 = 0$, and suppose the first busy period B_1 of $\{\zeta_i\}$ begins at step $\ell \ge 1$. Let D_1 be the duration of B_1. It is easy to check that, for any integer $y \ge 1$, the event $\{\zeta_i > 0 \text{ for all } i, \ell \le i \le \ell + y\}$ implies the event $\{\xi_{\ell+y} \ge \xi_\ell\}$. Busy periods are i.i.d. and $P(\xi_{\ell+y} \ge \xi_\ell)$ does not depend on ℓ, so

$$\begin{aligned} P(D > y) = P(D_1 > y) &\le P(\xi_{\ell+y} \ge \xi_\ell) \\ &\le P(\xi_y \ge 0) \ , \end{aligned} \tag{3.8}$$

By Lemma 3.1, we obtain that, for any $\epsilon > 0$, there exists an $\alpha > 0$ such that

$$P(\xi_y > (1-\epsilon)E[\xi_y]) = O(e^{\alpha E[\xi_y]}) \ , \tag{3.9}$$

with $E[\xi_y] = yE[U] < 0$. To see this, we need only observe that the U_i and hence ξ_y can be expressed as sums of independent 0-1 random variables. Put $\epsilon = 1$ in (3.9) and conclude that, for some $\alpha_1 > 0$,

$$P(\xi_y \ge 0) = O(e^{\alpha_1 y E[U]}) \ . \tag{3.10}$$

Together with (3.8), this proves the lemma. ∎

4. Admission Policy

The proof of Theorem 1.1 will use the admission policy of this section. Before presenting the policy, however, we will briefly review how it is applied in the general argument.

The proof of Theorem 1.1 estimates the expected value of the sum $S \equiv S(N,T)$ of waiting times under the greedy policy in an interval of length $T = \Theta(N^3)$, assuming that the state of the queues at the beginning of the interval is a sample from the stationary distribution. For convenience, we take $[0,T]$ as the interval. To make use of the estimate, observe that in the stationary regime, $E[Q_i(t)] = E[Q]$, so by (2.2) $E[S] = NTE[Q]$ and

$$E[Q] = \frac{E[S]}{NT} \ . \tag{4.1}$$

We will prove that, under the admission policy defined below, the sum $\tilde{S}$ of waiting times over $[0,T]$ satisfies $E[\tilde{S}] = O(N^3)$. By Theorem 2.1, $E[S] \le E[\tilde{S}]$, so substitution into (4.1) proves $E[Q] = O(1/N)$, since $T = \Theta(N^3)$. Then Theorem 1.1 is proved.

We now discuss the admission policy, algorithm A, shown in Fig. 3, previewing as we go along the properties of the algorithm that must be proved in the probabilistic analysis of the next section. The algorithm is based on various constants and structures determined by λ and μ, which we describe first. The algorithm takes as input an $\epsilon > 0$ such that $\mu(1-2\epsilon) > \lambda$, and reserves a sequence of $2\epsilon N$ cells of the ring as nearly equally spaced as possible. (The lengths of adjacent intervals between reserved cells differ by at most 1.) Call the odd numbered cells of this sequence *initialization* (I) cells, and the even numbered cells *clean-up* (CU) cells. *Regular* cells are those that are neither I nor CU cells.

To avoid trivialities and to simplify notation, we assume in what follows that $(2\epsilon)^{-1}$ and ϵN are integers. The reserved (I and CU) cells partition the $(1-2\epsilon)N$ regular cells into $2\epsilon N$ groups C_j, with $(2\epsilon)^{-1} - 1$ cells per group. We also define a partition of the queues into $2\epsilon N$ groups G_j, $1 \le j \le 2\epsilon N$, with $(2\epsilon)^{-1}$ per group. The index j is taken mod $2\epsilon N$ if $j > 2\epsilon N$.

Algorithm A determines the schedule over the time interval $[0, N + bN^3]$, which is partitioned into an initial block B_0 of N steps followed by N^2 blocks $B_1, \ldots, B_{N^2}$ of bN steps each. The parameter b must be chosen sufficiently small; the probabilistic analysis of Section 5 will give an upper bound in terms of μ and λ.

The algorithm is preceded by the following process: independently, at every cell and time step a mark ($\times$) is placed with probability μ/N; these marks are superposed on the input arrival pattern. If, by algorithm A a packet φ is placed in cell j at time t, then the first $\times$ after time t in column j signals the departure of φ from the ring (these particular $\times$'s correspond to bullets in Fig. 2). By the memoryless property of the geometric distribution of the times between successive $\times$'s in any column, this rule for determining departures yields geometric transit times on the ring, as desired.

With this set-up, the algorithm is as follows (see Fig. 3). First, the interval $[0, N]$ of B_0 is devoted solely to the accumulation of empty CU cells, to be used as described later, starting at time N. No admissions to the ring are scheduled during $[0, N]$. This is for convenience only; our asymptotic results would not change if such scheduling were allowed.

At time N, the algorithm partitions the empty CU cells into sequences σ_k, $1 \leq k \leq a \ln N$, as nearly equal in length as possible (see Step 1 in Fig. 3), for a constant a sufficiently large to be determined by the probabilistic analysis. Apart from their size and number, the sequences σ_k can be chosen arbitrarily from among the empty CU cells.

The remainder of Step 1 assigns I cells starting at time N to just those packets in the initial state plus those that arrived in $[0, N]$. The j-th I cell admits the packets in the ϵ^{-1} queues of $G_{2j-1} \cup G_{2j}$ at time N; it serves these queues in a round-robin sequence, i.e., a $(k+1)$-st packet from one of the queues is not admitted until at least k visits have been made to the other queues (admitting a packet at each visit if one is there).

Note that the I cells work in parallel with the other cells that serve the arrivals in $B_1, \ldots, B_{N^2}$. The probabilistic analysis will use elementary bounds to show that the expected total waiting time of packets served by I cells is negligible, i.e., $o(N^3)$.

Almost all of the arriving packets in the B_i, $1 \leq i \leq N^2$, are assigned by the iterations of Step 2 to regular cells during the interval $[N, N + bN^3]$ (see Fig. 3). Arrivals in B_i are assigned to cells at time $N + (i-1)bN$, $1 \leq i \leq N^2$. At this time, a regular cell is called *available* if its column segment in B_{i-1} has at least one $\times$, its column segments in B_{i-2} and B_{i-3} have no $\times$, and the cell has not already been assigned to an arrival in B_i (if $i = 2$, then the reference to B_{i-3} is omitted, and if $i = 1$, the references to both B_{i-2} and B_{i-3} are omitted). Examples are given in Fig. 4, which are referenced again at the end of this section. Step 2 scans the groups G_j in left-to-right order beginning with G_1. Assume for simplicity that the cell group C_j is lined up in front of the queues in G_j so that the last cell of C_j is in front of the last queue in G_j (recall that C_j has one fewer cell than G_j has queues). This is the alignment assumed in Step 2 of Fig. 3.

For $j = 1, \ldots, 2\epsilon N$ the arrivals as yet unassigned to $G_1, \ldots, G_j$ are assigned in any order to the available cells of C_{j+1} until either the former or latter set is empty, whichever occurs first. At the end of this process, there may still be unassigned arrivals in B_i; these are called *leftover* packets. Also, there may have been instances where an arrival was assigned to a cell more than bN cells (time units) away. These assignments are discarded and the corresponding packets are left unassigned throughout $[0, N + bN^3]$. The restriction to cells that are both available (in the above limited sense) and not too far from the arrivals assigned to them guarantees that algorithm A makes valid assignments. We will verify this fact after we describe the remainder of the algorithm.

The probabilistic analysis will show that, for each block B_i, the numbers of available cells in the C_j's is sufficiently large to ensure a $O(N)$ expected total waiting time for the arrivals assigned in Step 2. Then for all N^2 blocks, the total waiting time is $O(N^3)$, as desired. The analysis will then show that the assignment of an arrival to a cell more than bN columns away is so rare that its effect on total waiting time is negligible.

Finally, Step 3 of the algorithm takes care of leftover packets by assigning them to the cells of the sequences σ_k. These assignments are organized so that, for each $k = 1, \dots, a \ln N$, the leftover packets of B_k, $B_{k+a\ln N}$, $B_{k+2a\ln N}$, $\dots$ are all assigned to cells in the same sequence σ_k. Thus, for $r \geq 0$, the leftover packets in $B_{k+ra\ln N}, \dots, B_{k+(r+1)a\ln N-1}$ are served in parallel by disjoint regions of the ring.

The probabilistic analysis will show that, except for a negligible fraction of the leftover packets, all of those admitted by the cells of σ_k from $B_{k+ra\ln N}$ for any $r \geq 0$ will have departed when it is time to start admitting the arrivals in $B_{k+(r+1)a\ln N}$ into the cells of σ_k. In addition, the analysis will show that the expected total wait of the leftover packets in B_i is $O(N)$, and hence the expected total wait for leftovers from all N^2 blocks is $O(N^3)$, as desired.

It is easy to see that, if algorithm A always makes valid assignments (loads packets into empty cells), then it is indeed a valid admission policy; the (future) arrivals in B_i are known when assignments are made at the beginning of B_i, but knowledge of future departure times is not used at any point. (This is obvious for Steps 1 and 3; it is clear for Step 2 as well, since cell availability at the beginning of B_i depends only on departures times in $B_1 \cup \cdots \cup B_{i-1}$.)

It remains to verify that, under algorithm A, whenever an arrival reaches the cell to which it is assigned by the algorithm, the cell is empty. But suppose that cell j is the available cell assigned by Step 2(i) to arrival φ in B_i, and that it remains assigned to cell j

Algorithm A

Input: N, a, b, ϵ, an initial state and sets of arrivals and marks (×'s) over $[0, N + bN^3]$

1. (i) At time N, the empty CU cells are partitioned into sequences σ_k, $1 \leq k \leq a \ln N$, whose lengths differ by at most 1.

   (ii) For $j = 1, \ldots, \epsilon N$, the j-th I cell admits just those packets appearing in the ϵ^{-1} queues of $G_{2j-1} \cup G_{2j}$ at time N. For each j, these queues are served by a round-robin starting at time N.

For $i = 1, \ldots, N^2$ the following two steps are performed.

2. (i) Assume that the queues in G_j are aligned with the cells of C_j, at the time B_i begins. For $j = 1, \ldots, 2\epsilon N$, the as yet unassigned arrivals in the queues of $G_1, \ldots, G_j$ are assigned to the available cells in C_{j+1} until the former or the latter are exhausted, whichever occurs first ($C_{2\epsilon N+1} \equiv C_1$).

   (ii) Assignments just made that match an arrival to a cell more than bN columns (cells) away are removed.

3. Let integer k satisfy $1 \leq k \leq a$ and $i = ma + k$ for some integer $m \geq 0$. Then the leftover packets, if any, of B_i are admitted according to the greedy rule by the empty CU cells of σ_k; admissions stop when there are no more leftovers to admit or when the empty cells of σ_k have been exhausted, whichever occurs first.

Figure 3: An admission algorithm.

after Step 2(ii). (See Fig. 4 for examples.) Then the earliest that cell j can again become available occurs when assigning arrivals in B_{i+3}; no arrival of B_{i+1} or B_{i+2} can be assigned to cell j by the definition of cell availability and the fact that B_{i-1} has a × in column j. If cell j is indeed available during the scan of B_{i+3}, then there must be a × in B_{i+2}. This × must come after the admission of φ to cell j; otherwise the motion line of φ would span more than bN columns, and this would contradict Step 2(ii), where such assignments are removed. Thus, this × in B_{i+2} guarantees that any packet already in cell j will have departed before cell j is re-used for an arrival in B_{i+3} or some later block.

5. Proof of Theorem 1.1

Recall the general approach outlined at the beginning of the previous section: we prove that $E[\tilde{S}] = O(N^3)$, where $\tilde{S}$ is the sum of the waiting times in $[0, N+bN^3]$ under algorithm A. It is enough to show, as is done below, that the $O(N^3)$ bound holds for the packets considered by each of the three steps individually.

In what follows, when we say that an event occurs *with high probability*, we mean that it occurs with probability $1 - O(N^{-\gamma})$ where γ can be made as large as desired by a suitable choice of (usually hidden) constants. For example, by the geometric law for transit times V, we have

$$P(V \leq dN \log N) = 1 - (1 - \mu/N)^{dN \log N} \ ,$$

and so $P(V \leq dN \log N) \sim 1 - N^{-\mu d}$ as $N \to \infty$. Thus, we can say that transit times are $O(N \log N)$ with high probability; $\gamma = \mu d$ can be made as large as desired by increasing d. Note that m high-probability events occur *jointly* with high probability if m is at most some polynomial in N.

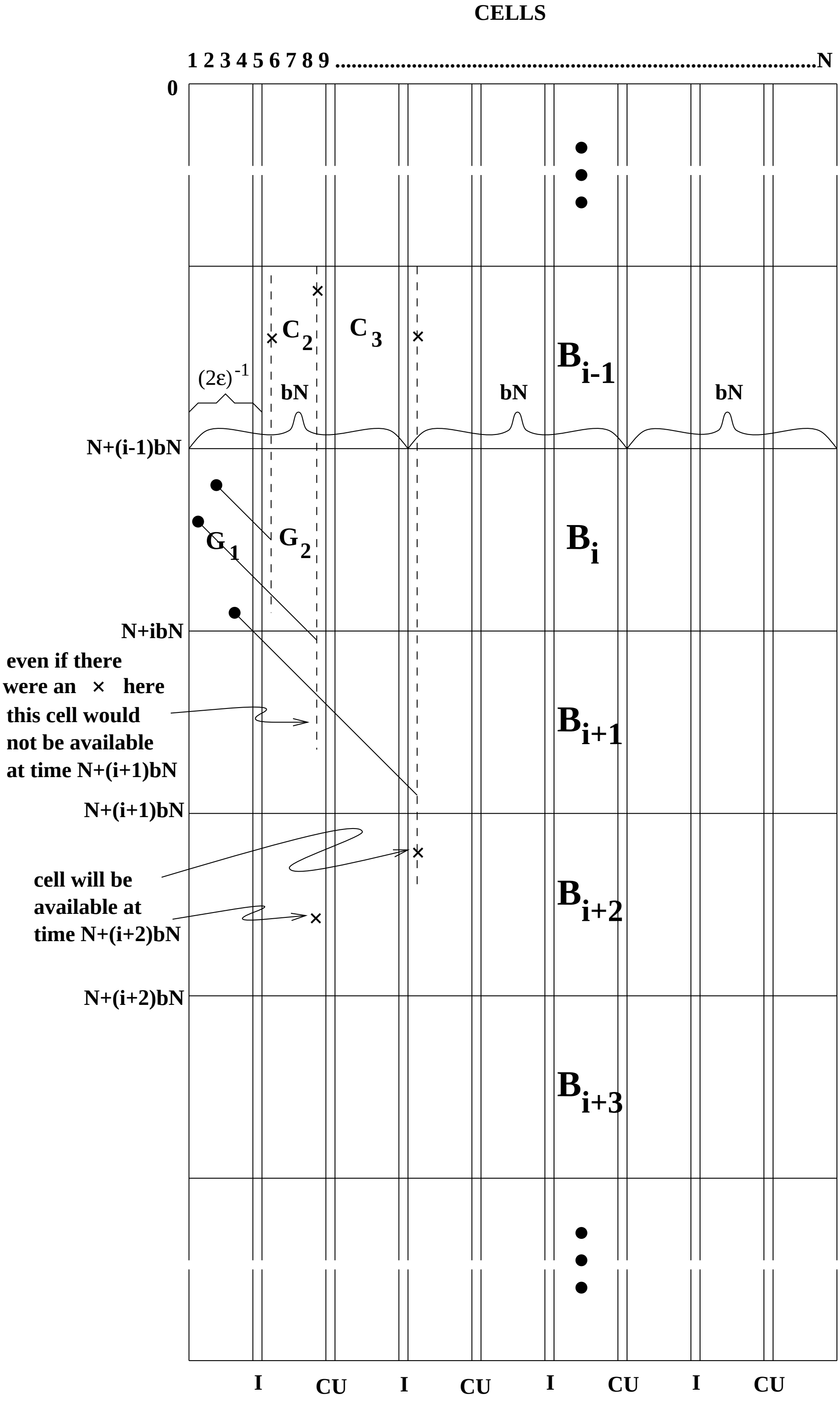
CELLS
1 2 3 4 5 6 7 8 9 ....N
0
C_2
C_3
$(2\varepsilon)^{-1}$
bN
bN
bN
B_{i-1}
N+(i-1)bN
G_2
G_1
B_i
N+ibN
even if there were an × here this cell would not be available at time N+(i+1)bN
B_{i+1}
N+(i+1)bN
cell will be available at time N+(i+2)bN
B_{i+2}
N+(i+2)bN
B_{i+3}
I
CU
I
CU
I
CU
I
CU

Figure 4: Admission Algorithm

Step 1.

Consider the i^{th} queue length $Q_i(N)$ at time N and recall that $Q_i(0)$ has the stationary distribution. Algorithm A admits no packets to the ring in $[0, N]$, so $Q_i(N)$ is $Q_i(0)$ plus the number of arrivals in $[0, N]$ at queue i. At time N, $Q_i^*(N)$ counts the packets waiting at time N in queue i plus the packet, if any, in the I cell that serves queue i in Step 1. We have $Q_i^*(N) \leq Q_i(N) + 1$, so by Lemma 2.1 and the geometric law of interarrival times, there exists an $\eta > 0$ such that $P(Q_i^*(N) \geq k+1) \leq P(Q_i(N) \geq k) = O(e^{-\eta k})$ independently for all queues. Let $\tilde{S}_i^{(1)}$ be the total waiting time of the packets counted by $Q_i(N)$, and let $\mathcal{C}$ be the joint event that (i) for some $c > 0$ the first $Q_i(N) \wedge c \ln N$ packets waiting in queue i at time N have $O(N \log N)$ transit times, and (ii) at time N the packet, if any, in the I cell serving queue i has $O(N \log N)$ remaining transit time. By the geometric law for transit times, $\mathcal{C}$ has high probability.

Since each I cell serves a constant number of queues in a round-robin sequence, the k^{th} packet in any queue must wait $k \cdot O(N \log N)$ time if $\mathcal{C}$ holds and $k \leq c \ln N$. Since no packet can wait more than $bN^3 + N$ time, we obtain the bound

$$\begin{aligned} E[\tilde{S}_i^{(1)} | \mathcal{C}] &= \sum_{1 \leq k \leq c \ln N} k \cdot O(N \log N) \quad + \quad O(N^3) \cdot E(Q_i(N) - c \ln N)^+ \\ &= O(N \log^3 N) + O(N^{-\eta c}) \cdot O(N^3) \\ &= O(N \log^3 N) \end{aligned}$$

by choosing c large enough. If $\mathcal{C}$ does not hold, we use the $O(N^3)$ trivial bound for all packets to obtain $E[\tilde{S}_i^{(1)} | \overline{\mathcal{C}}] = O(N^3)$, since $E[Q_i(N)] = O(1)$. But $\overline{\mathcal{C}}$ has low probability so $E[\tilde{S}_i^{(1)}] = O(N \log^3 N)$. Since $\mathcal{C}$ holds simultaneously for all i with high probability, we can conclude that

$$E[\tilde{S}^{(1)}] = N E[\tilde{S}_i^{(1)}] = O(N^2 \log^3 N) = O(N^3),$$

as desired.

Step 2.

We analyze the left-to-right scan of the sets G_j in B_i and C_j in B_{i-1}, and bound first the expected total waiting time of the packets that are assigned in Step 2(i). Define the Lindley process

$$\zeta_0 = 0, \quad \zeta_j = (\zeta_{j-1} + U_j)^+, \quad j = 1, \ldots, \epsilon N,$$

where $U_j = X_j - Y_j, X_j$ is the number of arrivals in G_j, and Y_j is the number of available columns in C_{j+1} at the start of the j-th iteration in Step 2. It is easy to see that, among the arrivals already scanned in $G_1, \ldots, G_j$, ζ_j gives the number as yet unassigned at the start of the $(j+1)^{\text{st}}$ iteration. Thus, $(2\epsilon)^{-1}(\zeta_1 + \cdots + \zeta_{2\epsilon N})$ is the cumulative waiting time $\tilde{S}_i^{(2)}$ of the arrivals assigned in B_i, not counting the times spent waiting by these arrivals in their initial and final blocks. The latter times are bounded by $2(2\epsilon)^{-1} = \epsilon^{-1}$, so

$$\tilde{S}_i^{(2)} \leq (2\epsilon)^{-1}(\zeta_1 + \cdots + \zeta_{2\epsilon N}) + \epsilon^{-1}(X_1 + \cdots + X_{2\epsilon N}).$$

But $E[X_j] = \lambda b (2\epsilon)^{-1}$, and if $E[U_j] < 0$ then $E[\zeta_j] = O(1)$, by Lemma 3.2, so that $E[\tilde{S}_i^{(2)}] = O(N)$ and hence the expected total wait of assigned packets summed over all i is $O(N^3)$, as desired. Thus, it remains to prove that $E[U_j] < 0$.

We need to verify that $P(A_k) > \lambda b/(1-2\epsilon)$, where A_k is the event that column k is available when assignments to arrivals in B_i begin; for then, since there are $(2\epsilon)^{-1} - 1$ columns in the C_j,

$$E[Y_j] > \frac{\lambda b}{1-2\epsilon}[(2\epsilon)^{-1} - 1] = (2\epsilon)^{-1}\lambda b = E[X_j],$$

and hence $E[U_j] < 0$. We consider B_i for $i \geq 4$; the cases $i = 1, 2, 3$ are similar. We need only observe that A_k holds if there is at least one $\times$ in a column of B_{i-1} and none in B_{i-2} and B_{i-3}, so

$$P(A_k) = [1 - (1 - \mu/N)^{bN}](1 - \mu/N)^{2bN} \sim (1 - e^{-\mu b})e^{-2\mu b}$$

as $N \to \infty$. Then for all N sufficiently large

$$b < \frac{1}{5}\frac{\mu(1-2\epsilon) - \lambda}{(1-2\epsilon)\mu^2}$$

is enough to ensure that $P(A_k) > \lambda b/(1-2\epsilon)$.

It remains to estimate the added total waiting time of the packets that were assigned but then unassigned in Step 2. But by Lemma 3.3 the probability that a packet is assigned to a $\times$ at least bN columns (and hence $\Omega(N)$ groups C_j) away is exponentially small in

N. It follows that the expected added total waiting time of such packets is $o(1)$ since the number and maximum wait of such packets are both bounded by polynomials in N. Thus, the expected total wait of all packets examined in Step 2 is $E[\tilde{S}^{(2)}] = O(N^3)$, as desired.

Step 3.

Let $k_i = (i-1) \bmod (a \ln N) + 1$, and note that, by Step 3, the leftover packets of B_i should go into the cells of σ_{k_i}.

We argue first that, by an application of Lemma 3.1, at time N there are at least $\frac{\epsilon}{2}(1-e^{-\mu})N$ empty CU cells with very high probability (i.e., with probability $1-O(e^{-\Omega(N)})$; thus, with very high probability, any existing leftover packets are assigned to the $a \log N$ sequences σ_k, which have $\Omega(N^\beta)$ cells each for every β, $0 < \beta < 1$.

For definiteness, choose $\beta = 1/2$ and let $\mathcal{E}_i$ be the event that the leftover packets of B_i number fewer than $N^{1/2}$ and each has a transit time at most $bN \cdot a \ln N - N$. In this event, the waiting time of each leftover packet is at most N and the leftover packets of B_i leave the CU cells of σ_{k_i} empty by the time the next set of leftover packets (those in $B_{i+a \ln N}$) have to be scheduled in the cells of σ_{k_i}. By (3.7) and the geometric law for transit times, $\mathcal{E}_i$ holds with high probability for all a large enough. There are only N^2 such events, so the combined event $\mathcal{E} = \bigcap_{i=0}^{N^2} \mathcal{E}_i$ also holds with high probability, where $\mathcal{E}_0$ is the event that there exist at least $\frac{\epsilon}{2}(1-e^{-\mu})N$ empty CU cells at time N.

Now suppose $\mathcal{E}$ holds. Then since a leftover packet waits at most N and there are N^2 blocks, the conditional expected total waiting time is $O(N^3)$ times the conditional expected number of leftover packets per block B_i, i.e., $O(N^3) \cdot E[\zeta_N|\mathcal{E}]$. But

$$E[\zeta_N|\mathcal{E}] = E[\zeta_N|\zeta_N \leq N^{1/2}] \leq E[\zeta_N] = O(1),$$

by Lemma 3.2, so the expected total waiting time of leftover packets is $O(N^3)$ when $\mathcal{E}$ holds.

Given that $\mathcal{E}$ does not hold, we use the trivial polynomial bounds $O(N^4)$ and $O(N^3)$ on the total number of leftover packets and the waiting time of each. Since $\mathcal{E}$ fails with low probability, a can be chosen large enough so that $P(\mathcal{E}) = 1 - O(N^{-4})$. Thus, the expected total waiting time of leftover packets is

$$O(N^3) + (1 - P(\mathcal{E}))O(N^7) = O(N^3)$$

and the theorem is proved. ∎

6. Final Remarks

A close look at the analysis in Section 5 shows that it is possible to prove a stronger version of Theorem 1.1 in which the dependence of the hidden multiplicative constant on λ and μ is specified: There exists a universal constant α such that, for N sufficiently large,

$$E[W] \leq \frac{\alpha}{(1-\rho)^2} \; . \tag{6.1}$$

The details of a proof of this result have been omitted because no new ideas are needed, and because the added clutter makes the proof significantly harder to follow. In broad outline, a proof can begin with the observation that if (6.1) can be proved for the expected waiting time $E[W^{(2)}]$ of packets assigned in Step 2, then changing only the constant α, it must also hold for $E[W]$. This is not difficult to verify using the probability bounds of Sections 2 and 3, and the arguments in Section 5.

It is then not difficult to verify that, within a constant factor independent of λ and μ, $E[W^{(2)}]$ is the expected waiting time in a G/G/1 queue with arrivals in each time slot having a binomial distribution with mean λb and service times having a geometric distribution with rate parameter $\mu' \approx (1-2\epsilon)(1-e^{-\mu b})e^{-2\mu b}$. For large N, the queue is asymptotically an M/G/1 queue, so by classical results, we get (Kleinrock (1975), Section 5.7)

$$E[W^{(2)}] = O(\frac{1}{(1-\rho')\mu'}),$$

where $\rho' = \lambda'/\mu'$ and $\lambda' = \lambda b$.

As in the analysis of Step 2 in Section 5, we again choose $b = \Theta(\frac{\mu-\lambda}{\mu^2})$, and so $1-\rho' = \Omega(1-\rho)$ and $\mu' = \Omega(\mu b) = \Omega(1-\rho)$. Thus, $E[W^{(2)}]$ and hence $E[W]$ has a bound of the form (6.1).

Asymptotics in N pose intriguing open problems for transit-time distributions other than the geometric. The uniform distribution on $\{1,\ldots,N-1\}$ is of particular interest; extensive simulations by Coffman et al. (1993) give convincing evidence that the bounds in Theorem 1.1 hold for this case as well, but no proof has yet been found.

Finally, keeping with our Markov arrival and transit-time assumptions, it would be interesting to study asymptotic behavior in the generalization of rings to toroidal arrays of processors (see Leighton (1990, 1992)). Much is known about regular (open) arrays, as can be seen from the recent work of Mitzenmacher (1994) and Kahale and Leighton (1995), who give references to the earlier work on this problem. But the analysis of toroidal arrays seems to require different methods.

Acknowledgment

We are grateful to M. Harchol-Balter for improvements to an early version of this paper, and to I. Telatar and A. Weiss for helpful discussions. The second author also thanks DIMACS for its hospitality and acknowledges its supporting agencies, the National Science Foundation under contract STC-91-19999 and the New Jersey Commission on Science and Technology.

References

Alon, N. and Spencer, J. H. (1991), *The Probabilistic Method*, Wiley & Sons, New York.

Asmussen, S. (1987), *Applied Probability and Queues*, Wiley & Sons, New York.

Barroso, L. A. and Dubois, M. (1992), "The Performance of Cache-Coherent Ring-Based Multiprocessors," *Proc. 20th Ann. Internat. ACM Symp. Comp. Arch.*, 268–277.

Billingsley, P. (1986), *Probability and Measure*, 2nd edition, Wiley & Sons, New York.

Coffman, E. G., Jr., Gilbert, E. N., Greenberg, A. G., Leighton, F. T., Robert, P. and Stolyar, A. L. (1995), "Queues Served by a Rotating Ring," *Stochastic Models*, **11**, 371–394.

Georgiadis, L., Szpankowski, W., and Tassiulas, L. (1994), "A Scheduling Policy with Maximal Stability Region for Ring Networks with Spatial Reuse," preprint. See also "Stability Analysis of Scheduling Policies in Ring Networks with Spatial Reuse," *Proc. 31st Ann. Allerton Conf. Comm. Cont. Comp.*, University of Illinois, Urbana.

Kahale, N. and Leighton, F. T. (1995), "Greedy Dynamic Routing on Arrays", *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM Press, 558 - 566.

Kleinrock, L. (1975), *Queueing Systems, Vol. I*, Wiley & Sons, New York.

Leighton, F. T. (1990), "Average Case Analysis of Greedy Routing Algorithms on Arrays," *Proc. 2nd Ann. ACM Symp. Parallel Algs. Arch.*, 2–10.

Leighton, F. T. (1992), *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA.

Mitzenmacher, M. (1994), "Bounds on the Greedy Algorithm for Array Networks," *Proc. 6th Ann. ACM Symp. Parallel Alg. Arch.*, 346–353.

Prabhu, N. U. (1965), *Stochastic Processes*, The Macmillan Co., New York.

Van Arem, B. and Van Doorn, E. A. (1990), "Analysis of a Queueing Model for Slotted Ring Networks," *Computer Networks and ISDN Systems*, **20**, 309–314.