Stochastic Systems
2016, Vol. 6, No. 1, 1–25
DOI: 10.1214/14-SSY151

ON QUEUE-SIZE SCALING FOR INPUT-QUEUED SWITCHES

By D. Shah, J. N. Tsitsiklis and Y. Zhong*

Massachusetts Institute of Technology

We study the optimal scaling of the expected total queue size in an $n \times n$ input-queued switch, as a function of the number of ports n and the load factor ρ, which has been conjectured to be $\Theta(n/(1-\rho))$ (cf. [15]). In a recent work [16], the validity of this conjecture has been established for the regime where $1-\rho = O(1/n^2)$. In this paper, we make further progress in the direction of this conjecture. We provide a new class of scheduling policies under which the expected total queue size scales as $O\big(n^{1.5}(1-\rho)^{-1}\log\big(1/(1-\rho)\big)\big)$ when $1-\rho = O(1/n)$. This is an improvement over the state of the art; for example, for $\rho = 1-1/n$ the best known bound was $O(n^3)$, while ours is $O(n^{2.5}\log n)$.

1. Introduction. An input-queued switch is a popular and commercially available architecture for scheduling data packets in an internet router. In general, an input-queued switch maintains a number of virtual queues to which packets arrive. Packets to be served at each time slot are selected according to a scheduling policy, subject to system constraints that specify which queues can be served simultaneously.

The input-queued switch model is an important example of so-called "stochastic processing networks," formalized by Harrison [5, 6], which have become a canonical model of a variety of dynamic resource allocation scenarios. While the most basic questions concerning throughput and stability[1] are relatively well-understood for general stochastic processing networks (see e.g., [10], [8], [7], [3], [18], [12], [19]), much less is known on the subject of

Received May 2014; revised 20 May 2015.

*This work was supported by NSF grants CCF-0728554 and CMMI-1234062. This research was performed while all authors were affiliated with the Laboratory for Information and Decision Systems as well as the Operations Research Center at MIT. The third author is currently with the IEOR department, at Columbia University. Current emails: {devavrat,jnt}@mit.edu, yz2561@columbia.edu.

MSC 2010 subject classifications: Primary 60K20, 68M12; secondary 68M20.

Keywords and phrases: Input-queued switch, queue-size scaling.

[1]Under the definition that we adopt, the system is *stable* if the expected queue sizes are bounded over time. Furthermore, a policy is *throughput optimal* if the system is stable whenever there exists some policy under which the system is stable.

more refined performance measures (e.g., results on the distribution and the moments of queue sizes), even for the special context of input-queued switches.

This paper contributes to the performance analysis of stochastic processing networks. It is motivated by the conjectures put forth in [15] on the optimal scaling of the expected total queue size in an $n \times n$ input-queued switch, as a function of the number of ports n and the load factor ρ. For certain limiting regimes, it was conjectured in [15] that the optimal scaling (that is, the scaling under an "optimal" policy) takes the form $\Theta\left(n/(1-\rho)\right)$. This is to be compared to available results that include an $O(n^2/(1-\rho))$ upper bound, which is a factor of $O(n)$ away from the conjectured scaling, and which is established for the so-called Maximum-Weight policy [14], [9], and an $O(n \log n/(1-\rho)^2)$ upper bound, which is a factor of $O(\log n/(1-\rho))$ away, and is achieved by a batching policy proposed in [13]. We also note an upper bound of $O(n^2/(1-\rho))$, achieved by a randomized policy [14], in the special case of uniform traffic. More recently, Shah et al. [16] proposed a policy that gives an upper bound of $\frac{n}{1-\rho} + n^3$, thus establishing the validity of the conjecture when $1-\rho = O(1/n^2)$.

In this paper, we focus on a different regime, where $1/n^2 \ll 1-\rho \leq 1/n$. In some sense, this is a more difficult regime to analyze, when compared to the regime where $1-\rho = O(1/n^2)$. This is because we consider a larger "gap" $1-\rho$, and so the heavy-traffic aspects of the system are less pronounced. This in turn means that various laws of large numbers (e.g., fluid or batching arguments) are less effective.

Concretely, we shall focus on the case $\rho = 1 - 1/f_n$, where $f_n \geq n$ for all n, and for n tending to infinity. When $f_n = n$, previous works give an upper bound $O(n^3)$ on the expected total queue size. In contrast, when $\rho = 1-1/n$, the conjectured optimal scaling $O\left(n/(1-\rho)\right)$ is of the form $O(n^2)$. It is then natural to ask whether this gap can be reduced, i.e., whether there exists a policy under which the expected total queue size is upper bounded by $O(n^\alpha)$, with $\alpha < 3$ (and ideally with $\alpha = 2$), when $\rho = 1 - 1/n$.

Our main contribution is a new policy that leads to an upper bound of $O\big(n^{1.5} f_n \log f_n\big)$, when $f_n \geq n$ and the arrival rates at the different queues are all equal. As a corollary, if $f_n = n$, the expected total queue size is upper bounded by $O(n^{2.5} \log n)$. This is the best known scaling with respect to n, when $\rho = 1 - 1/n$. (We also note that these scaling results can be extended to a class of arrival rates that is more general than the special case of equal rates.) While this is a significant improvement over existing bounds, we still believe that the right scaling (ignoring any poly-logarithmic factors) is $O(n^2)$. The best currently known scalings on the expected total queue size

TABLE 1
Best known scalings of the expected total queue size in various regimes. Here, ρ is the load factor and n is the number of input ports

Regime	Scaling	References
$\frac{1}{1-\rho} < n$	$O\left(\frac{n \log n}{(1-\rho)^2}\right)$	[13]
$\frac{1}{1-\rho} = n$	$O\left(n^{2.5} \log n\right)$	this work
$n \leq \frac{1}{1-\rho} < n^2$	$O\left(\frac{n^{1.5} \log n}{1-\rho}\right)$	this work
$\frac{1}{1-\rho} \geq n^2$	$\Theta\left(\frac{n}{1-\rho}\right)$	[16]

under various regimes, in an $n \times n$ input-queued switch, are summarized in Table 1.[2]

The policy that we propose is a variation of the standard batching policy. In the standard batching policy, time is divided into disjoint intervals or batches. Packets that arrive in a given batch are served only after the arrival of the entire batch. By choosing the batch length large enough (deterministically or randomly), the total number of arriving packets at each queue is close to its expected value and these packets can be served efficiently. In general, a longer batching interval improves efficiency, because the effect of random fluctuations is less pronounced, but on the other hand leads to larger delays and queue sizes. For this reason, a good batching policy, as for example in [13], selects the smallest possible batch length that will guarantee stability; in [13], this led to a bound of $O\left(\frac{n \log n}{(1-\rho)^2}\right)$ on the expected total queue size.

Given the stability requirement, we cannot hope to improve delay by reducing the batch length. On the other hand, the policy that we consider starts serving packets from a given batch a lot earlier, before the arrival of the entire batch. By starting to serve early, the expected delay (and hence queue size) is reduced. When the arrival rates at each queue are all equal, we

[2]After the submission of this paper, [11] established the heavy traffic optimality of the maximum weight policy, when arrival rates at the different queues are all equal. More specifically, using Lyapunov function drift techniques, [11] established non-asymptotic upper bounds on the steady-state total queue size, under uniform arrival rates and the maximum weight policy. As a corollary, for any fixed n, the steady-state total queue size has an upper bound of $O(n/(1-\rho))$, as $\rho \to 1$. However, in the regime where $1-\rho = O(1/n)$ and $n \to \infty$, which is considered in our paper, the non-asymptotic upper bound in [11] appears to be of a much higher order than $O(n^{2.5} \log n)$.

show that the arrival process has sufficient regularity at a time scale shorter than the batch length. Consequently, the policy can indeed start serving the arriving packets early, while making sure that the stochastic fluctuations lead to only a small number of unserved packets, which can be "cleared" efficiently at the end of the batch. The combination of these ideas results in substantial improvement over the standard batching policy.

A few remarks are in order regarding the proposed policy and its performance scaling. First, our policy relies on the assumption of uniform arrival rates. For a class of arrival rates that are more general than the special case of uniform rates (cf. Assumption 1 of Section 7), a similar performance bound of $O(n^{1.5} f_n \log f_n)$ can be achieved under a slight modification of the proposed policy, in the regime $\rho = 1 - 1/f_n$ and $f_n \geq n$. This modified policy and its performance scaling (Theorem 7.1) is presented in Section 7. Second, our policy (and its modification) makes use of the knowledge of the arrival rates. In contrast, some existing policies, such as the maximum weight policy or the one in [16], are based only on the observed system state (the queue sizes) and do not require knowledge of the arrival rates.

1.1. *Organization.* The rest of the paper is organized as follows. In Section 2, we describe the input-queued switch model. In Section 3, we state our main theorem. In Section 4, we introduce some preliminary facts and results, which will be used in later sections. In Section 5, we describe our policy. In Section 6, we provide the proof of the main theorem. The modified policy, for more general arrival rates is presented in Section 7. We conclude with some discussion in Section 8.

2. Input-queued switch model. An $n \times n$ input-queued switch has n input ports and n output ports. The switch operates in discrete time, indexed by $\tau \in \mathbb{N} = \{1, 2, \dots\}$. In each time slot, and for each port pair (i, j), a unit-sized packet may arrive at input port i destined for output port j, according to an exogenous arrival process. Let $A_{i,j}(\tau)$ denote the cumulative number of such arriving packets during time slots $1, \dots, \tau$. We assume that the processes $A_{i,j}(\cdot)$ are independent for different pairs (i, j). Furthermore, for every input-output pair (i, j), $\{A_{i,j}(\tau) - A_{i,j}(\tau - 1)\}_{\tau \in \mathbb{N}}$ is a Bernoulli process with parameter ρ/n, with the convention that $A_{i,j}(0) = 0$. In particular,

$$\mathbb{E}[A_{i,j}(\tau)] = \frac{\rho}{n}\tau, \quad \text{for all } i, j, \text{ and all } \tau \geq 1.$$

We are only interested in systems that can be made stable under a suitable policy, and for this reason, we assume that $\rho < 1$, i.e., that the system is

underloaded. Furthermore, we consider a system load ρ of the form $\rho = 1 - 1/f_n$, where the sequence $\{f_n\}$ satisfies $f_n \geq n$ for all n.

For every input-output pair (i, j), the associated arriving packets are stored in separate queues, so that we have a total of n^2 queues. Let $Q_{i,j}(\tau)$ be the number of packets waiting at input port i, destined for output port j, at the beginning of time slot τ.

At each time slot, the switch can transmit a number of packets from input ports to output ports, subject to the following two constraints: (i) each input port can transmit at most one packet; and, (ii) each output port can receive at most one packet. In other words, the actions of a switch at a particular time slot constitute a *matching* between input and output ports.

A matching, or *schedule*, can be described by an array $\boldsymbol{\sigma} \in \{0,1\}^{n\times n}$, where $\sigma_{i,j} = 1$ if input port i is matched to output port j, and $\sigma_{i,j} = 0$ otherwise. Thus, at any given time, the set of all feasible schedules is

$$\mathcal{S} = \Big\{\boldsymbol{\sigma} \in \{0,1\}^{n\times n} : \sum_k \sigma_{i,k} \leq 1,\ \sum_k \sigma_{k,j} \leq 1,\ \forall\ i, j\Big\}. \tag{1}$$

A scheduling policy (or simply *policy*) is a rule that, at any given time τ, chooses a schedule $\boldsymbol{\sigma}(\tau) = [\sigma_{i,j}(\tau)] \in \mathcal{S}$, based on the past history and the current queue sizes $Q_{i,j}(\tau)$. If $\sigma_{i,j}(\tau) = 1$ and $Q_{i,j}(\tau) > 0$, then one packet is removed from the queue associated with the pair (i, j).

Regarding the details of the model, we adopt the following timing conventions. At the beginning of time slot τ, the queue sizes $Q_{i,j}(\tau)$ are observed by the policy. The schedule $\boldsymbol{\sigma}(\tau)$ is applied in the middle of the time slot. Finally, at the end of the time slot, new arrivals happen. Mathematically, for all i, j, and $\tau \in \mathbb{N}$, we have

$$Q_{i,j}(\tau + 1) = Q_{i,j}(\tau) - \sigma_{i,j}(\tau)\mathbf{1}_{\{Q_{i,j}(\tau)>0\}} + A_{i,j}(\tau) - A_{i,j}(\tau-1), \tag{2}$$

where for event B, $\mathbf{1}_B$ is the associated indicator function. We assume throughout the paper that the system starts empty, i.e., $Q_{i,j}(1) = 0$, for all i, j.

Summing Eq. (2) over time and using the assumption $Q_{i,j}(1) = 0$, we get the following equivalent expression, for $\tau \in \mathbb{N}$:

$$Q_{i,j}(\tau+1) = A_{i,j}(\tau) - \sum_{t=1}^{\tau} \sigma_{i,j}(t)\mathbf{1}_{\{Q_{i,j}(t)>0\}}. \tag{3}$$

We define

$$S_{i,j}(\tau) = \sum_{t=1}^{\tau} \sigma_{i,j}(t)\mathbf{1}_{\{Q_{i,j}(t)>0\}},$$

so that (3) reduces to

$$Q_{i,j}(\tau+1) = A_{i,j}(\tau) - S_{i,j}(\tau).$$

We call $S_{i,j}(\tau)$ the *actual* service received by queue (i,j) during the first τ time slots. Note that $S_{i,j}(\tau)$ may be different from $\sum_{t=1}^{\tau} \sigma_{i,j}(t)$, which is the cumulative service *offered* to queue (i,j) during the first τ slots.

3. Main Result. The main result of this paper is as follows.

THEOREM 3.1. *Consider an $n \times n$ input-queued switch in which the arrival processes are independent Bernoulli processes with a common arrival rate ρ/n, where $\rho = 1 - 1/f_n$ and $f_n \geq n$. For any n, there exists a scheduling policy under which the expected total queue size is upper bounded by $cn^{1.5} f_n \log f_n$. That is,*

$$\sum_{i,j=1}^{n} \mathbb{E}[Q_{i,j}(\tau)] \leq cn^{1.5} f_n \log f_n, \qquad \text{for all } \tau,$$

where c is a constant that does not depend on n.

COROLLARY 3.2. *Consider the setup in Theorem 3.1, with $f_n = n$. For any n, there exists a scheduling policy under which the expected total queue size is upper bounded by $cn^{2.5} \log n$. That is,*

$$\sum_{i,j=1}^{n} \mathbb{E}[Q_{i,j}(\tau)] \leq cn^{2.5} \log n, \qquad \text{for all } \tau,$$

where c is a constant that does not depend on n.

Let us remark here that we only prove Theorem 3.1 for all sufficiently large n. The validity of the theorem for smaller n is guaranteed by considering an arbitrary stabilizing policy (e.g., the maximum weight policy) and letting c be large enough so that we have an upper bound to the expected total queue size under that policy.

4. Preliminaries. Here we state some facts that will be used in our subsequent analysis.

Concentration Inequalities. We will use the following tail bounds for binomial random variables (adapted from Theorem 2.4 in [2]).

THEOREM 4.1. *Let $X_1, X_2, \ldots, X_m$ be independent and identically distributed Bernoulli random variables, with*

$$\mathbb{P}(X_i = 1) = p, \quad \text{and} \quad \mathbb{P}(X_i = 0) = 1 - p,$$

for $i = 1, 2, \ldots, m$. Let $X = \sum_{i=1}^{m} X_i$, so that $\mathbb{E}[X] = mp$. Then, for any $x > 0$, we have

$$\text{(4) (Lower tail)} \qquad \mathbb{P}(X \leq \mathbb{E}[X] - x) \leq \exp\left\{-\frac{x^2}{2\mathbb{E}[X]}\right\},$$

$$\text{(5) (Upper tail)} \qquad \mathbb{P}(X \geq \mathbb{E}[X] + x) \leq \exp\left\{-\frac{x^2}{2(\mathbb{E}[X] + x/3)}\right\}.$$

Kingman Bound for the discrete-time $G/G/1$ Queue. Consider a discrete-time $G/G/1$ queueing system. More precisely, let $X(\tau)$ be the number of packets that arrive during time slot τ, let $Y(\tau)$ be the number of packets that can be served during slot τ, and let $Z(\tau)$ be the queue size at the beginning of time slot τ. Suppose that the $X(\tau)$ are i.i.d. across time, and so are the $Y(\tau)$. Furthermore, the processes $X(\cdot)$ and $Y(\cdot)$ are independent. The queueing dynamics are given by

$$Z(\tau + 1) = \max\{0, Z(\tau) + X(\tau) - Y(\tau)\}. \tag{6}$$

Note that the timing convention in (6) is different from that of the queueing dynamics (2) of the switch model: with Eq. (6), during each slot, arrivals take place before any service. This timing convention will be used later on to analyze the so-called *backlogged* packets under the policy that we propose, which evolve similar to (6) (cf. (13)).

Let $\lambda = \mathbb{E}[X(\tau)]$, $m_{2x} = \mathbb{E}[X^2(\tau)]$, $\mu = \mathbb{E}[Y(\tau)]$, and $m_{2y} = \mathbb{E}[Y^2(\tau)]$. Suppose that $\lambda < \mu$. The following bound is proved in [17] (Theorem 3.4.2), using a standard argument based on a quadratic Lyapunov function.

THEOREM 4.2 (Discrete-time Kingman bound). *Suppose that $Z(1) = 0$ and that $\lambda < \mu$. Then,*

$$\mathbb{E}[Z(\tau)] \leq \frac{m_{2x} + m_{2y} - 2\lambda\mu}{2(\mu - \lambda)}, \qquad \text{for all } \tau. \tag{7}$$

In fact, the above theorem is proved in [17] for the expected queue size in steady state. However, since we assume that $Z(1) = 0$, a standard coupling argument shows that the same bound holds for $\mathbb{E}[Z(\tau)]$ at any time τ.

Optimal Clearing Policy. Similar to [13], we will use the concept of the *minimum clearance time* of a queue matrix. Consider a certain queue matrix $[Q_{i,j}]_{i,j=1}^n$, where $Q_{i,j}$ denotes the number of packets at input port i destined for output port j. Suppose that no new packets arrive, and that the goal is to simply clear all packets present in the system, in the least possible amount of time, using only feasible schedules/matchings. We call this minimal required time the *minimum clearance time* of the given queue matrix, and we denote it by L. Then, L is characterized exactly as follows.

THEOREM 4.3. *Let $[Q_{i,j}]_{i,j=1}^n$ be a queue matrix. Let*

$$R_i = \sum_{j=1}^{n} Q_{i,j} \qquad \text{and} \qquad C_j = \sum_{i=1}^{n} Q_{i,j}$$

be the ith row sum and the jth column sum, respectively. Then, the minimum clearance time, L, is equal to the largest of the row and column sums:

$$L = \max\left\{\max_i R_i, \max_j C_j\right\}. \tag{8}$$

The proof of Theorem 4.3 is a simple modification of the proof of Theorem 5.1.9 in [4].

Note that in each time slot at most one packet can depart from each input/output port, and therefore each R_i and C_j is decreased by at most 1. Thus, the minimum clearance time cannot be smaller than the right-hand side of (8). Theorem 4.3 states that there actually exists an *optimal clearing policy* that clears all packets within exactly $\max\{\max_i R_i, \max_j C_j\}$ time slots.

5. Policy Description. To describe our policy, we introduce three parameters, b_n, d_n, and s_n, which depend on the number of ports, n. These parameters specify the lengths of certain time intervals, which, in turn, delineate the different phases of the policy. They are given by[3]

$$b_n = c_b f_n^2 \log f_n, \tag{9}$$

$$d_n = c_d \sqrt{n} f_n \log f_n, \tag{10}$$

$$s_n = \rho b_n + \sqrt{c_s b_n \log f_n}. \tag{11}$$

[3]We will treat these parameters as if they were guaranteed to be integers. Rounding them up or down to a nearest integer would overburden our notation but would have no effect on our order-of-magnitude estimates.

Without loss of generality, we will always assume that $n \geq 3$, so that $\log f_n > 1$. Here c_b, c_d, and c_s are positive constants (independent of n) that will be appropriately chosen. As will be seen in the course of the proof, it suffices to choose them so that

$$c_b > c_s, \qquad c_d^2 \geq 640 c_b, \qquad c_d > c_b, \qquad c_s \geq 30, \tag{12}$$

and which we henceforth assume. We note that the above inequalities do not necessarily lead to the best choices for these constants but they are imposed in order to simplify the details of the proof. In the rest of the paper, and to avoid overburdening notation, we will suppress the subscript n from the parameters b_n, d_n, and s_n, and write them as b, s, and n.

For an $n \times n$ input-queued switch, we introduce n particular schedules $\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}, \ldots, \boldsymbol{\sigma}^{(n)}$. For $u \in \{1, 2, \ldots, n\}$, $\boldsymbol{\sigma}^{(u)}$ is defined by

$$\sigma_{i,j}^{(u)} = \begin{cases} 1, & \text{if } j = i + u - 1 \pmod{n}, \\ 0, & \text{otherwise.} \end{cases}$$

To illustrate, when $n = 3$, the schedules $\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}$, and $\boldsymbol{\sigma}^{(3)}$ are given by

$$\boldsymbol{\sigma}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\sigma}^{(2)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\sigma}^{(3)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Note that

$$\boldsymbol{\sigma}^{(1)} + \boldsymbol{\sigma}^{(2)} + \cdots + \boldsymbol{\sigma}^{(n)} = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix},$$

the $n \times n$ matrix of all 1s.

We now proceed with the description of the policy. Time is divided into consecutive intervals, which we call *arrival periods*, of length b. For $k = 0, 1, 2, \ldots$, the kth arrival period consists of slots $kb+1, kb+2, \ldots, (k+1)b$. Arrivals that occur during the kth arrival period are said to belong to the kth *batch*.

The general idea behind the policy is as follows. The policy aims to serve all of the packets in the kth batch during the kth *service period*, of length b, which is offset from the arrival period by a delay of d. Thus, the kth service period consists of time slots $kb + d + 1, \ldots, (k + 1)b + d$. If the policy does not succeed in serving all of the packets in the kth batch, the unserved packets will be considered *backlogged* and will be handled together with newly arriving packets from subsequent batches, in subsequent service

periods. As it will turn out, however, the number of backlogged packets will be zero, with high probability.

We now continue with a precise description, by considering what happens during the kth service period. Note that the time slots $kb+1, \ldots, kb+d$ do not belong to the kth service period. Packets from the kth batch will accumulate during these time slots, but none of them will be served. At the beginning of the kth service period (the beginning of time slot $bk+d+1$), we may have some backlogged packets from previous service periods, and we denote their number by B_k. We assume that $B_0 = 0$.

The kth service period consists of three phases, which are described below and are illustrated in Fig. 1.

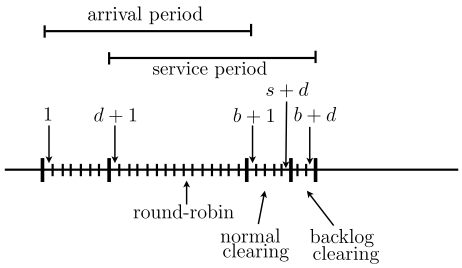

FIG 1. *Illustration of a typical arrival period and the phases of a service period. Slots are numbered consecutively, starting with the first slot of the arrival period.*

1. The first $b-d$ slots of the kth service period, namely, slots $kb+d+1, \ldots, (k+1)b$, comprise a *round-robin phase*: we cycle through the schedules $\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}, \ldots, \boldsymbol{\sigma}^{(n)}$ in a round-robin manner. However, during this phase, we do not serve any of the backlogged packets; we only serve packets that belong to the kth batch.[4]
2. The next $\ell = d+s-b$ slots, namely slots $(k+1)b+1, \ldots, kb+d+s$, comprise the kth *normal clearing phase*. Similar to the round-robin phase, we do not serve any backlogged packets during this phase. Furthermore, even though packets from the $(k+1)$st batch may have started to arrive, we do not serve any of them. By the beginning of this phase, all of the arrivals from the kth batch have already arrived. Some of them have already been served during the round-robin phase. To those that remain, we apply the optimal clearing policy described

[4]This particular choice introduces some inefficiency, because offered service will be wasted whenever a queue has backlogged packets but no packets that belong to the kth batch. However, this choice simplifies our analysis and makes little actual difference, because the number of backlogged packets is zero with high probability.

earlier; cf. Theorem 4.3. However, there is a possibility that the phase terminates before we succeed in serving all of the remaining packets from the kth batch. Let U_k be the number of the packets from the kth batch that were left unserved during this phase. These U_k packets are considered backlogged and are added to the backlog B_k from earlier periods.

3. The last $r = b - s$ slots, namely slots $kb + d + s + 1, \ldots, (k+1)b + d$, comprise the kth *backlog clearing phase*. During this phase, we serve backlogged packets using some arbitrary policy. The only requirement is that the policy serve at least one packet at each slot that a backlogged packet is available. However, we do not serve any of the newly arrived packets from the $(k+1)$st batch. Any backlogged packets that are not served during this phase remain backlogged and comprise the number B_{k+1} of backlogged packets at the beginning of the next service period. Since at least one backlogged packet is served (whenever available) during each one of these r slots, and since there are no additions to the backlog during this phase, we have

$$B_{k+1} \le \max\{0, B_k + U_k - r\}, \qquad k = 0, 1, \ldots \tag{13}$$

The total length of the three phases is

$$(b - d) + (d + s - b) + (b - s) = b,$$

so that the length of a service period is equal to the length of an arrival period. However, before continuing, we need to make sure that the duration of each phase is a positive number, so that the policy is well-defined. This is accomplished in the next two lemmas, which also provide order of magnitude information on the durations of these phases.

LEMMA 5.1. *The length $r = b - s$ of the backlog clearing phase satisfies*

$$r = c_r f_n \log f_n,$$

where $c_r = c_b - \sqrt{c_s c_b} > 0$. In particular, when n is large enough, we have $r \ge 1$.

PROOF. Using the assumption $\rho = 1 - 1/f_n$, we have $(1-\rho)b = b/f_n = c_b f_n \log f_n$. We then obtain

$$\begin{aligned} b - s &= b - \rho b - \sqrt{c_s b \log f_n} \\ &= c_b f_n \log f_n - \sqrt{c_s c_b f_n^2 \log^2 f_n} \end{aligned}$$

$$\begin{aligned} &= (c_b - \sqrt{c_s c_b}) f_n \log f_n \\ &= c_r f_n \log f_n. \end{aligned}$$

The fact that $c_r > 0$ follows from our assumption in Eq. (12). □

LEMMA 5.2. *The length $\ell = d + s - b$ of the normal clearing phase satisfies*

$$\ell \geq c_\ell \sqrt{n} f_n \log f_n,$$

where $c_\ell = c_d - c_r > 0$. In particular, when n is large enough, we have $\ell \geq 1$.

PROOF. Recall that $r = b - s$. It follows that

$$\begin{aligned} \ell &= d + s - b \\ &= d - r \\ &= c_d \sqrt{n} f_n \log f_n - c_r f_n \log f_n \\ &\geq (c_d - c_r) \sqrt{n} f_n \log f_n \\ &= c_\ell \sqrt{n} f_n \log f_n. \end{aligned}$$

Note that $c_r < c_b < c_d$ (cf. Lemma 5.1 and Eq. (12)), which implies that $c_\ell > 0$. □

6. Policy Analysis. The performance analysis of the proposed policy involves the following line of argument for what happens during the kth arrival and service period.

(a) In the first d slots of the kth arrival period, we have an expected number $O(nd)$ of arrivals.
(b) With high probability, at every time slot during the round-robin phase, there is a positive number of packets from the kth arrival batch at each queue; cf. Lemmas 6.1 and 6.2. Therefore, offered service is never wasted. In particular, at least as many packets are served as they arrive (in the expected value sense), and the total queue size does not grow.
(c) With high probability, all of the packets from the kth batch that are in queue at the beginning of the normal clearing phase get cleared and therefore the number U_k of newly backlogged packets is zero; cf. Lemma 6.4.
(d) The number B_k of backlogged packets evolves similar to a discrete-time G/D/1 queue; cf. Eq. (13). Because U_k is zero with high probability, the Kingman bound (Theorem 4.2) implies that the expected number of backlogged packets, at any time, is small; cf. Lemma 6.5.

The above steps, when translated into precise bounds on queue sizes, will lead to an $O(nd)$ bound on the expected total queue size at any time.

6.1. *No waste during the round-robin phase.* In this subsection, we establish that during the round-robin phase, every queue contains a nonzero number of packets from the current arrival batch, with high probability. We first introduce some convenient notation. We will use the variable $t \in \{1, \ldots, b+1\}$ to index the b slots of the kth arrival period together with the first slot of the subsequent normal clearing phase. For $t \in \{1, \ldots, b\}$, we let $A^k_{i,j}(t)$ be the number of arrivals to the (i,j)th queue during the first t time slots of the kth arrival period; these are the time slots $kb+1, kb+2, \ldots, kb+t$. Similarly, for $t \in \{1, \ldots, b\}$, we let $S^k_{i,j}(t)$ be the number of packets that arrive to queue (i,j) during the kth arrival period and get served during the first t time slots of the kth arrival period. Finally, for $t \in \{1, \ldots, b+1\}$, we let $Q^k_{i,j}(t)$ be the number of packets from the kth arrival batch that are in queue (i,j) at the beginning of the tth slot of the kth arrival period. With these definitions, we have,

$$Q^k_{i,j}(t+1) = A^k_{i,j}(t) - S^k_{i,j}(t), \qquad t = 1, \ldots, b. \tag{14}$$

We are interested in conditions under which no offered service is wasted during the round-robin phase. Equivalently, we are interested in conditions under which all queues have a positive number of packets from the kth batch. Note that the round-robin phase involves slots for which $t \in \{d+1, \ldots, b\}$. We have the following observation on the queue sizes at the beginning of these slots.

LEMMA 6.1. *Suppose that* $t \in \{d, \ldots, b-1\}$ *and that*

$$A^k_{i,j}(t) > \frac{t-d}{n} + 1.$$

Then, $Q^k_{i,j}(t+1) > 0$.

PROOF. Note that that for the first d time slots, packets from the kth batch do not receive any service. Starting from the $(d+1)$st slot, we are in the round-robin phase, and queue (i,j) is offered service once every n slots. Therefore,

$$S^k_{i,j}(t) \leq \left\lceil \frac{t-d}{n} \right\rceil < \frac{t-d}{n} + 1 < A^k_{i,j}(t).$$

The result follows from Eq. (14). □

The previous lemma highlights the importance of the events $A^k_{i,j}(t) > (t-d)/n + 1$. We will show that the complements of these events have,

collectively, small probability. To this effect, let $W^k_{i,j}(t)$ be the event defined by

$$W^k_{i,j}(t) = \Big\{ A^k_{i,j}(t) \leq \frac{t-d}{n} + 1 \Big\}, \qquad t = d, \ldots, b-1.$$

Let also W^k be the union of these events, over all queues, and over all indices t that are relevant to the round-robin phase:

$$W^k = \bigcup_{i=1}^{n} \bigcup_{j=1}^{n} \bigcup_{t=d}^{b-1} W^k_{i,j}(t).$$

LEMMA 6.2. *For n sufficiently large, we have*

$$\mathbb{P}(W^k) \leq \frac{1}{2f_n^{13}}, \qquad \textit{for all } k.$$

PROOF. Let us fix some (i,j) and some $t \in \{d, \ldots, b-1\}$. Note that $\mathbb{E}\big[A^k_{i,j}(t)\big] = \rho t/n$. Therefore, the event $W^k_{i,j}(t)$ is the same as the event

$$\Big\{ A^k_{i,j}(t) \leq \mathbb{E}\big[A^k_{i,j}(t)\big] - \frac{\rho t}{n} + \frac{t-d}{n} + 1 \Big\},$$

which is of the form

$$\Big\{ A^k_{i,j}(t) \leq \mathbb{E}\big[A^k_{i,j}(t)\big] - x \Big\},$$

where

$$\begin{aligned} x &= \frac{\rho t}{n} - \frac{t-d}{n} - 1 \\ &= \frac{\rho(t-d)}{n} - \frac{t-d}{n} + \frac{\rho d}{n} - 1 \\ &= -(1-\rho)\frac{t-d}{n} + \frac{\rho d}{n} - 1. \end{aligned}$$

Using the facts $t - d \leq b$ and $1 - \rho = 1/f_n$, the first term on the right-hand side is bounded above (in absolute value) by $b/(nf_n)$. For the second term, we use the facts $\rho = 1 - (1/f_n)$, $f_n \geq n \geq 2$, to obtain $\rho \geq 1/2$. Therefore,

$$\begin{aligned} x &\geq -\frac{b}{nf_n} + \frac{d}{2n} - 1 \\ &= \frac{1}{n}\Big((c_d/2)\sqrt{n} f_n \log f_n - c_b f_n \log f_n - n \Big) \\ &\geq \frac{1}{n}\Big((c_d/2)\sqrt{n} f_n \log f_n - (c_b+1) f_n \log f_n \Big). \end{aligned}$$

Now, for n large enough, we have $c_b + 1 \le (c_d/4)\sqrt{n}$, and this implies that

$$x \ge \frac{1}{n} \cdot \frac{c_d}{4} \cdot \sqrt{n} f_n \log f_n = \frac{c_d f_n \log f_n}{4\sqrt{n}}. \tag{15}$$

Using Eq. (4) (the lower tail bound in Theorem 4.1), we have

$$\mathbb{P}\big(W_{i,j}^k(t)\big) = \mathbb{P}\Big(A_{i,j}^k(t) \le \mathbb{E}\big[A_{i,j}^k(t)\big] - x\Big) \le \exp\Big\{-\frac{x^2}{2\mathbb{E}[A_{i,j}^k(t)]}\Big\}.$$

We note that $\mathbb{E}[A_{i,j}^k(t)] = \rho t/n \le b/n = c_b f_n^2 (\log f_n)/n$. Using also Eq. (15), we obtain

$$\frac{x^2}{2\mathbb{E}[A_{i,j}^k(t)]} \ge \frac{c_d^2 f_n^2 \log^2 f_n}{16n} \cdot \frac{1}{2c_b f_n^2 (\log f_n)/n} = \frac{c_d^2}{32 c_b} \log f_n \ge 20 \log f_n,$$

where the last inequality follows from our assumption that $c_d^2 \ge 640 c_b$; cf. Eq. (12). Consequently,

$$\mathbb{P}\big(W_{i,j}^k(t)\big) \le \exp\{-20 \log f_n\} = \frac{1}{f_n^{20}} \le \frac{1}{2 f_n^{19}}.$$

The event W^k is the union of $n^2(b-d)$ events $W_{i,j}^k(t)$. We note that

$$n^2(b-d) \le n^2 b \le f_n^2 c_b f_n^2 \log f_n \le f_n^6, \tag{16}$$

as long as n is large enough so that $c_b \le f_n$. Therefore, using the union bound

$$\mathbb{P}(W^k) \le n^2(b-d)\frac{1}{2 f_n^{19}} \le \frac{f_n^6}{2 f_n^{19}} = \frac{1}{2 f_n^{13}}.$$

□

6.2. *The probability of no new backlog.* In this subsection we show that U_k, the additional backlog generated during the kth service period, is zero with high probability. Our analysis builds on an upper bound on the probability that the number of packets in the kth batch that are associated with a particular port is appreciably larger than its expected value. Towards this purpose, we define the row and column sums for the arrivals in the kth batch:

$$R_i^k = \sum_j A_{i,j}^k(b), \qquad C_j^k = \sum_i A_{i,j}^k(b).$$

We also define the events

$$F_i^k = \{R_i^k > s\}, \qquad G_j^k = \{C_j^k > s\},$$

and

$$H^k = \big(F_1^k \cup \cdots \cup F_n^k\big) \cup \big(G_1^k \cup \cdots \cup G_n^k\big).$$

In what follows, we first show that the event H^k has low probability. We then show that if neither of the events W^k or H^k occurs (which has high probability), then U_k is equal to zero.

LEMMA 6.3. *For n sufficiently large, we have*

$$\mathbb{P}(H^k) \le \frac{1}{2f_n^{13}}, \qquad \textit{for all } k.$$

PROOF. Let us focus on the event $F_1^k = \{R_1^k > s\}$; the argument for other events F_i^k or G_j^k is identical. Note that $\mathbb{E}[R_1^k] = \rho b$. We have, using Eq. (5) (the upper tail bound in Theorem 4.1) in the last step,

$$\begin{aligned}
\mathbb{P}(R_1^k > s) &= \mathbb{P}\big(R_1^k > \rho b + \sqrt{c_s b \log f_n}\big) \\
&= \mathbb{P}\big(R_1^k > \mathbb{E}[R_1^k] + \sqrt{c_s b \log f_n}\big) \\
&\le \exp\Big\{ - \frac{c_s b \log f_n}{2(\rho b + x/3)} \Big\},
\end{aligned}$$

where $x = \sqrt{c_s b \log f_n}$. Notice that

$$\rho b + \frac{x}{3} \le \rho b + x = \rho b + \sqrt{c_s b \log f_n} = s \le b,$$

where the last inequality follows from Lemma 5.1. Therefore, when $n \ge 4$,

$$\mathbb{P}(R_1^k > s) \le \exp\Big\{ - \frac{c_s b \log f_n}{2b} \Big\} = \frac{1}{f_n^{c_s/2}} \le \frac{1}{4f_n^{14}},$$

where the last inequality follows from our assumption that $c_s \ge 30$; cf. Eq. (12). The event H^k is the union of $2n$ events, each with probability bounded above by $1/(4f_n^{14})$. Using the union bound and the assumption $n \le f_n$, we obtain $\mathbb{P}(H^k) \le 1/(2f_n^{13})$. □

LEMMA 6.4.

(a) Consider a sample path under which neither W^k nor H^k occurs. Then, $U_k = 0$.
(b) We have $\mathbb{P}(U_k > 0) \le 1/f_n^{13}$.
(c) For every sample path, we have $U_k \le n^2 b$.

PROOF. (a) We assume that neither W^k nor H^k occurs. Using Eq. (14), the queue sizes (where we only count packets from the kth batch) at the beginning of the normal clearing period are equal to

$$Q^k_{i,j}(b+1) = A^k_{i,j}(b) - S^k_{i,j}(b). \tag{17}$$

Let

$$\hat{R}^k_i = \sum_j Q^k_{i,j}(b+1), \qquad \hat{C}^k_j = \sum_i Q^k_{i,j}(b+1).$$

Now consider a fixed i. Note that the schedules $\boldsymbol{\sigma}^{(m)}$ used during the round-robin phase have the property $\sum_j \sigma^{(m)}_{i,j} = 1$; that is, each input port is offered exactly one unit of service at each time slot. Furthermore, since event W^k does not occur, Lemma 6.1 implies that all queues are positive at the beginning of each slot of the round-robin phase; that is, $Q^k_{ij}(t+1) > 0$, for $t = d, \ldots, b-1$. Therefore, the offered service is never wasted during the $b-d$ slots of the round-robin phase. It follows that the total actual service at input port i during the round-robin phase is exactly $b-d$:

$$\sum_j S^k_{i,j}(b) = b - d.$$

Furthermore, since event H^k does not occur, we have $R^k_i \le s$. Recalling the definition $R^k_i = \sum_j A^k_{i,j}(b)$, and by summing both sides of Eq. (17) over all j, we obtain

$$\hat{R}^k_i = R^k_i - \sum_j S^k_{i,j}(b) \le s - (b-d) = \ell,$$

where $\ell = d + s - b$ is the length of the normal clearing phase. By a similar argument, we obtain that $\hat{C}^k_j \le \ell$, for all j. It then follows from Theorem 4.3 that all the packets (from the kth arrival batch) will be cleared during the normal clearing phase, and $U_k = 0$.

(b) If $U_k > 0$, then, by part (a), it must be that either event W^k or H^k occurs. The result follows because the probability of each one of these two events is upper bounded by $1/(2f^{13}_n)$ (Lemmas 6.2 and 6.3).

(c) The number of packets from the kth batch that can get backlogged can be no more than the total number of arrivals in the kth batch. Since each queue (n^2 of them) receives at most one packet at each time slot (b slots), the total number cannot exceed $n^2 b$.

□

6.3. *Backlog analysis.* We are now in a position to show that the expected backlog is very small.

LEMMA 6.5. *Assuming that n is sufficiently large, we have that $\mathbb{E}[B_k] \leq 1$, for all k.*

PROOF. Using Eq. (13), the backlog satisfies

$$B_{k+1} \leq \max\{0, B_k + U_k - r\} \leq \max\{0, B_k + U_k - 1\},$$

where the last inequality follows from Lemma 5.1. Let us define a sequence $\hat{B}_k$ with the recursion $\hat{B}_0 = 0$ and

$$\hat{B}_{k+1} = \max\{0, \hat{B}_k + U_k - 1\}.$$

We then have $B_k \leq \hat{B}_k$, so it suffices to derive an upper bound on $\mathbb{E}[\hat{B}_k]$.

We use the discrete-time Kingman bound (Theorem 4.2), where we identify $Z(\tau)$ with $\hat{B}_k$, $X(\tau)$ with U_k, and $Y(\tau)$ with 1. Using the notation in Theorem 4.2, we have $\mu = 1$, and $m_{2y} = 1$. Furthermore, as in Eq. (16), we have $n^2 b \leq f_n^6$ for sufficiently large n. Using Lemma 6.4,

$$\lambda = \mathbb{E}[U_k] \leq f_n^6 \cdot \mathbb{P}(U_k > 0) \leq f_n^6 \cdot \frac{1}{f_n^{13}} = \frac{1}{f_n^7},$$

and

$$m_{2x} = \mathbb{E}[U_k^2] \leq f_n^{12} \cdot \mathbb{P}(U_k > 0) = f_n^{12} \cdot \frac{1}{f_n^{13}} = \frac{1}{f_n}.$$

Then, using the bound in (7), we have

$$\mathbb{E}[B_k] \leq \mathbb{E}[\hat{B}_k] \leq \frac{m_{2x} + m_{2y}}{2(\mu - \lambda)} \leq \frac{f_n^{-1} + 1}{2(1 - f_n^{-7})}.$$

As n increases, the right-hand side converges to $1/2$ and is therefore bounded above by 1 when n is sufficiently large. □

6.4. *Queue size analysis.* In this subsection, we prove Theorem 3.1, the main result of the paper. Toward this end, we show that at any time, the sum of the queue sizes is of order $O(nd)$. We fix some time τ and consider two cases, depending on whether this time belongs to a round-robin phase or not.

Queue sizes during the round-robin phase. Suppose that τ satisfies $kb + d + 1 \leq \tau \leq (k+1)b$, so that τ belongs to the round-robin phase of the

kth service period, and let us look at the queue size $Q_{i,j}(\tau+1)$. This queue size may include some packets that arrived during earlier arrival periods and that were backlogged; their total expected number (summed over all i and j) is $\mathbb{E}[B_k] \le 1$.

Let us now turn our attention to packets that belong to the kth batch. Recall that the number of such packets in queue (i,j) at the beginning of the $(t+1)$st slot (equivalently, the end of the tth slot) of the kth arrival period is denoted by $Q^k_{i,j}(t+1)$. For $t = d+1,\dots,b$, we have, as in Eq. (14),

$$Q^k_{i,j}(t+1) = A^k_{i,j}(t) - S^k_{i,j}(t),$$

and

$$\sum_{i,j} \mathbb{E}[Q^k_{i,j}(t+1)] = n\rho t - \mathbb{E}\Big[\sum_{i,j} S^k_{i,j}(t)\Big].$$

By the same argument as in the proof of Lemma 6.4(a), if event W^k does not occur, the service during the round-robin phase is never wasted: a total of n packets are served at each time, and for $t = d+1,\dots,b$, a total of $n(t-d)$ packets are served by the tth slot of the kth arrival period. Using also the inequality (cf. Lemma 6.2)

$$1 - \mathbb{P}(W^k) \ge 1 - \frac{1}{2f_n^{13}} \ge 1 - \frac{1}{f_n} = \rho,$$

we obtain

$$\mathbb{E}\Big[\sum_{i,j} S^k_{i,j}(t)\Big] \ge n(t-d)\big(1-\mathbb{P}(W^k)\big) \ge n\rho(t-d).$$

Therefore,

$$\sum_{i,j} \mathbb{E}[Q^k_{i,j}(t+1)] \le n\rho t - n\rho(t-d) = n\rho d \le nd, \qquad t = d+1,\dots,b. \tag{18}$$

which is an upper bound of the desired form.

Queue sizes outside the round-robin phase. Suppose now that τ satisfies $(k+1)b+1 \le \tau \le (k+1)b+d$, so that τ belongs to one of the last two phases of the kth service period, and let us look again at the queue size $Q_{i,j}(\tau+1)$. As before, we may have some backlogged packets. These are either packets backlogged during the current period (the kth one) or in previous periods. Their total expected number (summed over all i and j) at any time in this range is upper bounded by $\mathbb{E}[B_k + U_k] \le 2$.

Let us now turn our attention to packets that belong to the kth batch. Since there are no further arrivals from the kth batch from slot $(k+1)b+1$

onwards, the number of such packets is largest at the beginning of slot $(k+1)b+1$. Their expected value at that time satisfies

$$\sum_{i,j} \mathbb{E}\big[Q_{i,j}^{k}(b+1)\big] \le nd,$$

where in the inequality we used Eq. (18) with $t = b$.

Finally, we need to account for arrivals that belong to the $(k+1)$st arrival batch. The total number of such accumulated arrivals is largest when we consider the largest value of τ, namely, $\tau = (k+1)b + d$. By that time, we have had a total of d slots of the $(k+1)$st arrival period, and a total expected number of arrivals equal to ρnd, which is bounded above by nd.

Putting together all of the bounds that we have developed, we see that at any time, the expected total number of packets is bounded above by $2nd + 2 \le 3nd$. This being true for all sufficiently large n, establishes Theorem 3.1.

7. Extension to More General Arrival Rates. As mentioned in the Introduction, it is possible to modify the policy presented in Section 5 so that it can accommodate non-uniform arrival rates, while achieving a similar performance bound of $O(n^{1.5} f_n \log f_n)$ (see Theorem 7.1). We first define the class of arrival rates that we consider in Section 7.1, and then state the performance properties of the modified policy in Section 7.2. The modified policy is described in Section 7.3. We provide a sketch of the performance analysis of the modified policy in Section 7.4, which is very similar to the analysis presented in Section 6.

7.1. *Assumption on arrival rates.* As in Section 2, we assume that the arrival processes $A_{i,j}(\cdot)$ are independent for different pairs (i,j), and that for each input-output pair (i,j), $\{A_{i,j}(\tau) - A_{i,j}(\tau-1)\}_{\tau\in\mathbb{N}}$ is a Bernoulli process with parameter $\lambda_{i,j}$. For each i, define $\tilde{\rho}^i = \sum_j \lambda_{i,j}$ to be the load on input port i; and for each j, define $\tilde{\rho}_j = \sum_i \lambda_{i,j}$ to be the load on output port j. Let $\rho = \max\big\{\max_i \tilde{\rho}^i, \max_j \tilde{\rho}_j\big\}$ be the system load. As in Section 2, we assume that $\rho = 1 - 1/f_n$ with $f_n \ge n$ for all n. Furthermore, we assume that the arrival rates are not heavily skewed, in the following sense.

ASSUMPTION 1. *There exists a positive constant c_0 such that for all n, and for all $i, j \in \{1, 2, \ldots, n\}$,*

$$\lambda_{i,j} \ge \frac{c_0}{n}.$$

Let us remark that when the arrival rates are uniform, i.e., when $\lambda_{i,j} = \rho/n$ for all i and j, and when $\rho = 1 - f_n$ with $f_n \ge n$, Assumption 1 holds with

$c_0 = 1/2$. As a result, the modified policy to be described in Section 7.3 can be applied to the case of uniform arrival rates, and achieves a performance bound of the same order as the policy in Section 5. In this paper, we have chosen to present the policy of Section 5 in detail, because its analysis is cleaner and better conveys the essential ideas, while only providing a sketch for the general case, in this section. We also note that the class of arrival rates considered in this section is fairly restrictive, excluding many instances of non-uniform arrival rates. We leave the investigation of general non-uniform arrival rates as future work.

7.2. *Main result.* The following theorem and corollary are extensions of Theorem 3.1 and Corollary 3.2, respectively, for arrival rates that satisfy Assumption 1.

THEOREM 7.1. *Consider an $n \times n$ input-queued switch in which the arrival processes are independent Bernoulli processes with arrival rates satisfying Assumption 1, and where $\rho = 1 - 1/f_n$ and $f_n \geq n$. For any n, there exists a scheduling policy under which the expected total queue size is upper bounded by $\tilde{c}n^{1.5} f_n \log f_n$. That is,*

$$\sum_{i,j=1}^{n} \mathbb{E}[Q_{i,j}(\tau)] \leq \tilde{c}n^{1.5} f_n \log f_n, \qquad \text{for all } \tau,$$

where $\tilde{c}$ is a constant that only depends on c_0, and which, in particular, does not depend on n.

COROLLARY 7.2. *Consider the setup in Theorem 7.1, with $f_n = n$. For any n, there exists a scheduling policy under which the expected total queue size is upper bounded by $\tilde{c}n^{2.5} \log n$. That is,*

$$\sum_{i,j=1}^{n} \mathbb{E}[Q_{i,j}(\tau)] \leq \tilde{c}n^{2.5} \log n, \qquad \text{for all } \tau,$$

where $\tilde{c}$ is a constant that only depends on c_0, and which, in particular, does not depend on n.

7.3. *Policy description.* The modified policy, for the case of arrival rates that satisfy Assumption 1, is very similar to the one described in Section 5. Indeed, the modified policy also consists of *service periods*, which are offset from the corresponding *arrival periods* by a delay of $\tilde{d}$ (see (20)). Each service period of the modified policy also consists of three phases, which we

call the *randomized service phase*, the *normal clearing phase*, and the *backlog clearing phase*. We now proceed with further details.

Similar to the description in Section 5, we introduce three parameters $\tilde{b}$, $\tilde{d}$, and $\tilde{s}$, given by

$$\tilde{b} = \tilde{c}_b f_n^2 \log f_n, \tag{19}$$

$$\tilde{d} = \tilde{c}_d \sqrt{n} f_n \log f_n, \tag{20}$$

$$\tilde{s} = \rho \tilde{b} + 2\sqrt{\tilde{c}_s \tilde{b} \log f_n}. \tag{21}$$

The positive constants $\tilde{c}_b$, $\tilde{c}_d$ and $\tilde{c}_s$ are chosen so that

$$\tilde{c}_d c_0 > 15\tilde{c}_b, \qquad \tilde{c}_s \geq 40, \qquad \tilde{c}_d > \tilde{c}_b > 4\tilde{c}_s. \tag{22}$$

The kth arrival period consists of slots $k\tilde{b}+1, k\tilde{b}+2, \ldots, (k+1)\tilde{b}$, and the kth service period consists of slots $k\tilde{b}+\tilde{d}+1, k\tilde{b}+\tilde{d}+2, \ldots, (k+1)\tilde{b}+\tilde{d}$. The kth service period is divided into three phases: a randomized service phase consisting of the first $\tilde{b}-\tilde{d}$ slots, a normal clearing phase consisting of the next $\tilde{\ell} = \tilde{d}+\tilde{s}-\tilde{b}$ slots, and a backlog clearing phase consisting of the last $\tilde{r} = \tilde{b}-\tilde{s}$ slots. Similar to the proofs of Lemmas 5.1 and 5.2, it can be shown that the quantities $\tilde{r}$ and $\tilde{\ell}$ are positive for sufficiently large n, so that the phases are well-defined. The descriptions of the normal clearing and the backlog clearing phases are exactly the same as those of the policy described in Section 5, so we focus on the description of the randomized service phase.

The randomized service phase. For each pair (i,j), define

$$\tilde{\lambda}_{i,j} = \lambda_{i,j} + \frac{1}{nf_n}.$$

Then, it is easy to see that

$$\sum_j \tilde{\lambda}_{i,j} \leq 1 \text{ for all } i, \quad \text{and} \quad \sum_i \tilde{\lambda}_{i,j} \leq 1 \text{ for all } j.$$

Let $\tilde{\Lambda} = (\tilde{\lambda}_{i,j})_{i,j=1}^n$ be the matrix with entries $\tilde{\lambda}_{i,j}$. By the Birkhoff-von Neumann theorem [1], there exist schedules $\boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots, \boldsymbol{\pi}^{(m)} \in \mathcal{S}$ (recall the definition of the set $\mathcal{S}$ of feasible schedules in (1)) and positive constants $\alpha_1, \alpha_2, \ldots, \alpha_m$ such that

$$\tilde{\Lambda} = \alpha_1 \boldsymbol{\pi}^{(1)} + \alpha_2 \boldsymbol{\pi}^{(2)} + \ldots + \alpha_m \boldsymbol{\pi}^{(m)}, \quad \text{and} \quad \alpha_1 + \cdots + \alpha_m = 1.$$

At each time slot during the randomized service phase, schedule $\boldsymbol{\pi}^{(u)}$ is chosen with probability α_u, $u = 1, 2, \ldots, m$, independently from other time

slots. In particular, this implies that for each pair (i, j), input i is matched to output j with probability $\tilde{\lambda}_{i,j}$.

Similar to the round-robin phase of the policy in Section 5, we do not serve any of the backlogged packets during the randomized service phase, and only serve packets that belong to the kth batch. This completes the description of the randomized service phase.

7.4. *Sketch of policy analysis.* The analysis of the modified policy follows the same line of argument as the analysis presented in Section 6: (a) in the first $\tilde{d}$ slots of the kth arrival period, we have an expected number $O(n\tilde{d})$ of arrivals; (b) offered service is never wasted with high probability, during the randomized service phase; (c) with high probability, all packets from the kth batch get cleared at the end of the normal clearing phase; and (d) the expected number of backlogged packets at any time is small.

Let us remark that Assumption 1 is used to show statement (b): with high probability (w.h.p.), offered service is never wasted during the randomized service phase. To see this, let us use the notation $A^k_{i,j}(t)$ and $S^k_{i,j}(t)$ from Section 6.1. Then, under the modified policy, w.h.p., for all $t \in \{\tilde{d}, \ldots, \tilde{b}-1\}$ and for all pairs (i, j),

$$A^k_{i,j}(t) > \lambda_{i,j}t - \sqrt{40\lambda_{i,j}\tilde{b}\log f_n}, \text{ and } S^k_{i,j}(t) < \tilde{\lambda}_{i,j}(t-\tilde{d}) + \sqrt{40\tilde{\lambda}_{i,j}\tilde{b}\log f_n}.$$

Offered service is never wasted as long as $S^k_{i,j}(t) < A^k_{i,j}(t)$, for all $t \in \{\tilde{d}, \ldots, \tilde{b}-1\}$, which happens, w.h.p., if

$$\lambda_{i,j}t - \sqrt{40\lambda_{i,j}\tilde{b}\log f_n} > \tilde{\lambda}_{i,j}(t-\tilde{d}) + \sqrt{40\tilde{\lambda}_{i,j}\tilde{b}\log f_n}.$$

The choice of constants satisfying (22) and Assumption 1 are used to establish the preceding inequality. Intuitively, Assumption 1 ensures that enough packets arrive to each queue during the first $\tilde{d}$ time slots of the kth arrival period, which is required to offset the stochastic fluctuations during the randomized service phase. We omit further details.

8. Discussion. We presented a novel scheduling policy for an $n \times n$ input-queued switch. In the regime where the system load satisfies $\rho = 1-1/n$, and the arrival rates at the different queues are all equal, our policy achieves an upper bound of order $O(n^{2.5}\log n)$ on the expected total queue size, a substantial improvement upon earlier upper bounds, all of which were of order $O(n^3)$, ignoring poly-logarithmic dependence on n. Our policy is of the batching type. However, instead of waiting until an entire batch has arrived, our policy only waits for enough arrivals to take place for the

system to exhibit a desired level of regularity, and then starts serving the batch. This idea may be of independent interest.

Our policy uses detailed knowledge of the arrival statistics, and is heavily dependent on the fact that all arrival rates are the same. While it is possible to relax the assumption of uniform arrival rates to some extent, the description and analysis of similar policies for arbitrary arrival rates (within the regime considered in this paper), are likely to be more involved.

Finally, for the regime where $\rho \approx 1 - 1/n$, there is a $\Omega(n^2)$ lower bound on the expected total queue size under any policy (see [15]), whereas our upper bound is of order $O(n^{2.5} \log n)$. It is an interesting open question whether this gap between the upper and lower bound can be closed. Our policy uses a prespecified set of schedules (round-robin or randomized schedules) until the entire batch has arrived and then uses an "adaptive" sequence of schedules to clear remaining packets after the end of the batch. Within the class of policies of this type, with perhaps different choices of the parameters involved, it appears to be impossible to obtain an upper bound of $O(n^\alpha)$ for $\alpha < 2.5$. Thus, in order to come closer to the $\Omega(n^2)$ lower bound, we will have to use an adaptive sequence of schedules early on, before the entire batch has arrived. In fact, if one were to achieve an upper bound close to $O(n^2)$, we would have an approximately constant expected number of packets in each queue. This means that with positive probability, many of the queues will be empty. Therefore, an elaborate policy would be needed to avoid offering service to empty queues and thus avoid queue buildup. But the analysis of such elaborate policies appears to be a difficult challenge.

REFERENCES

[1] G. Birkhoff. Tres observaciones sobre el algebra lineal. Uniu. Nac. Tkcumdn Rev. Ser. A **5**, 147–151 (1946) MR0020547

[2] F. Chung. *Complex graphs and networks.* American Mathematical Society (2006) MR2248695

[3] J. G. Dai and B. Prabhakar. The throughput of switches with and without speed-up. Proceedings of IEEE Infocom, pp. 556–564 (2000)

[4] M. Jr. Hall. *Combinatorial theory.* Wiley-Interscience, 2nd edition (1998)

[5] J. M. Harrison. Brownian models of open processing networks: canonical representation of workload. The Annals of Applied Probability **10**, 75–103 (2000). URL http://projecteuclid.org/euclid.aoap/1019737665. Also see [6] MR1765204

[6] J. M. Harrison. Correction to [5]. The Annals of Applied Probability **13**, 390–393 (2003) MR1952004

[7] F. P. Kelly and R. J. Williams. Fluid model for a network operating under a fair bandwidth-sharing policy. The Annals of Applied Probability **14**, 1055–1083 (2004) MR2071416

[8] I. Keslassy and N. McKeown. Analysis of scheduling algorithms that provide 100% throughput in input-queued switches. Proceedings of Allerton Conference on Communication, Control and Computing (2001)

[9] E. Leonardi, M. Mellia, F. Neri and M. A. Marsan. Bounds on average delays and queue size averages and variances in input queued cell-based switches. Proceedings of IEEE Infocom, pp. 1095–1103 (2001)
[10] W. Lin and J. G. Dai. Maximum pressure policies in stochastic processing networks Operations Research, **53**, 197–218 (2005) MR2131925
[11] S. T. Maguluri and R. Srikant. Heavy-traffic behavior of the MaxWeight algorithm in a switch with uniform traffic. Preprint available at `http://arxiv.org/pdf/1503.05872v1.pdf`, April 2015.
[12] N. McKeown, V. Anantharam and J. Walrand. Achieving 100% throughput in an input-queued switch. Proceedings of IEEE Infocom, pp. 296–302 (1996)
[13] M. Neely, E. Modiano and Y. S. Cheng. Logarithmic delay for $n \times n$ packet switches under the cross-bar constraint. IEEE/ACM Transactions on Networking **15**(3) (2007)
[14] D. Shah and M. Kopikare. Delay bounds for the approximate Maximum Weight matching algorithm for input queued switches. Proceedings of IEEE Infocom (2002)
[15] D. Shah, J. N. Tsitsiklis and Y. Zhong. Optimal scaling of average queue sizes in an input-queued switch: an open problem. Queueing Systems **68**(3-4), 375–384 (2011) MR2834209
[16] D. Shah, N. Walton and Y. Zhong. Optimal queue-size scaling in switched networks. Accepted to appear in the Annals of Applied Probability (2014) MR3262502
[17] R. Srikant and L. Ying. *Communication networks: An optimization, control and stochastic networks perspective.* Cambridge University Press (2014) MR3202391
[18] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Transactions on Automatic Control **37**, 1936–1948 (1992) MR1200609
[19] G. de Veciana, T. Lee and T. Konstantopoulos. Stability and performance analysis of networks supporting elastic services. IEEE/ACM Transactions on Networking **9**(1), 2–14 (2001)

D. Shah
E-mail: devavrat@mit.edu

J. N. Tsitsiklis
E-mail: jnt@mit.edu

Y. Zhong
E-mail: yz2561@columbia.edu