Online Supplement
to
Creating Work Breaks From Available Idleness

Xu Sun and Ward Whitt

Department of Industrial Engineering and Operations Research, Columbia University
New York, NY, 10027

June 30, 2017

1 Overview

In this online supplement to the main paper we primarily provide more information about our simulation experiments. First, in §2 we describe how we implemented the D_1, SISF and D_2 server-assignment rules. In §3 we provide additional simulation results for the work-conserving D_1 server-assignment rule. In §4, we present additional simulation results for the D_2 rule, in particular, for a large service system with $n = 1000$. In §5, we examine another way to announce work breaks from the available idleness. We refer to this rule as $LISF - D_2$. Finally, in §6 we elaborate on the existence of a critical threshold for D_1 in the fluid model, supplementing the discussion in §3.1 of the main paper, and presenting another derivation of equation (3.12) in the main paper.

2 Implementing the Server-Assignment Rules in the Simulations

In this section we describe how we implemented the D_1, SISF and D_2 server assignment rules in our simulation experiments. We consider these in turn in §2.1, §2.2 and §2.3. The rules D_1 and SISF are defined in equations (2.3) and (2.3) in §2.2 of the main paper. The Rule D_2 is defined at the beginning of §5 of the main paper.

2.1 Implementing the D_1 Server-Assignment Rule

Let any idle time greater than or equal to θ be called an (unannounced) *break*. Following an object-oriented-programming approach, we treat each server as an "object" from a "server class;" e.g., see Horstmann (2002). Each server contains three "instance variables," namely its identity number, service completion time and break end time. To implement D_1 in a virtual environment, we maintain for each busy server a service-completion time; this value is infinity by default for idle servers. Similarly, for each idle server we maintain a break end time by acting as if its current idle period will eventually develop into a break; this value is infinity by default for busy servers.

We conduct a discrete-event simulation in which no change in the system occurs between consecutive events. Thus the simulation jumps in time from one event to the next. For D_1, three types of events can happen: (i) customer arrival, (ii) customer departure and (iii) end (completion) of break. The algorithm maintains (a) a FIFO queue for

waiting customers, (b) a high-priority-queue (HPQ) containing all servers whose elapsed idle time exceeds θ and (c) a sorted list L with all the server other than those in the HPQ in the order of increasing ages.

At each arrival epoch, we look for idle servers in the HPQ. If any, assign the server at the head of the HPQ, reset its age to zero and move the server to the head of the list L. Otherwise, we scan through the list L to find an idle server with the shortest age. We make assignment if there exists such a server in L; otherwise the customer is put in queue.

For the selected idle server, the algorithm generates a service requirement S from the service-time distribution and resets its service completion time to $t + S$. Then we find the minimum service-completion time among all busy servers and update the departure time accordingly. Searching for the closest service completion time can be costly if the number of servers n is large. To accelerate the search, we arrange all busy servers in a binary heap where the root node is the server with the minimum service completion time. Computationally this is efficient, because it takes $O(1)$ operations to extract the minimum and $O(\log(n))$ operations to restore the heap structure as new elements enter.

At each departure epoch, we first look for customers in queue. The server gets assigned if the queue is nonempty. Otherwise the server becomes idle. At this time, we reset its service completion time to infinity, set the break-end time to $t + \theta$ and update the closest break-end time.

At the end of a break, we move the idle server to the back of the HPQ. That prevents a break from being much greater than θ, because we first assign idle servers from the HPQ.

2.2 Implementing the $SISF$ Server-Assignment Rule

To implement $SISF$, we stipulate that each server belongs to one of the three places: (i) the busy-server pool (BSP), (ii) the low-priority-queue (LPQ) for assignment or (iii) the high-priority-queue (HPQ) for assignment. For each busy server, we maintain the time for the current task to complete and set this value to infinity for idle servers. Similarly, for each idle server in the LPQ we maintain a break end time by assuming that its current idle period would eventually develop into an idle period of length θ; we set this value to infinity for busy servers as well as (idle) servers in the HPQ.

At each arrival epoch, we look to see if the HPQ is empty; if it is nonempty, we assign the server at the head of the HPQ. If the HPQ is empty, we look for idle servers in the LPQ and assign a server (if any) from the back of the LPQ. We use the first-in first-out (FIFO) discipline in the HPQ, but the last-in first-out (LIFO) discipline in the LPQ. Because the HPQ is FIFO, we use a circular array to implement the HPQ. The LPQ is a LIFO queue except that when a break finishes the server at the head of the LPQ joins the back of the HPQ (at this time we reset its break end time to infinity). We therefore use a linked-list to efficiently implement the LPQ.

Once a server gets assigned, we put the server into the BSP and attach to it a service completion time by sampling from the service-time distribution. Here we calculate (update) the minimum service completion time and let it be the time of next departure. Again we use a binary heap as we did for rule D_1 to speed up the searches for the minimum service completion time among all busy servers.

If no customers wait in queue, each customer departure is followed by a removal of a server from the BSP and its joining the LPQ. At this time, we set its service completion time to infinity and schedule its next long-idle-period

end time.

2.3 Implementing the D_2 Server-Assignment Rule

We consider five types of events: customer arrival, customer departure (service completion), due for a break, announced break completion and unannounced break completion. We first explain how to treat the control parameter τ with $\eta = \infty$, so it plays no role. Afterwards, we discuss the modifications to include η.

At each customer arrival epoch, we look for available servers. If any, assign the server with the shortest age. For the selected idle server, the algorithm generates a service requirement S from the service-time distribution and resets its service completion time to $t + S$. Then we find the minimum service-completion time among all busy servers and update the departure time accordingly. If there are no servers available, the arriving customer waits in queue.

At each customer departure epoch, we look for customers in queue. If there is a customer waiting, assign the newly-available server to the head-of-line customer. Otherwise, let the newly-available server either become idle or start a break depending on whether or not a high priority designation (to be explained momentarily) was given. If a high priority designation was given, the break is announced and the server is off duty and not available to provide service for a duration θ after that time. Otherwise it remains idle.

At each break due time (when a server's age reaches τ), if the server is busy, then we give the server a high priority designation indicating that its next idle period will be replaced by an announced break. If the server is idle, then the server starts an announced break and goes off duty for the duration θ. (The elapse idle time at the time of the break is not included in the break, and is counted as part of the total idle time.)

At each announced-break-end time, we first reset the server's age to zero. We assign this newly-available server to a customer if the queue is not empty. Otherwise, the newly-available server stays idle.

At each unannounced-break-end time, we reset the server's age to zero. At this time the queue must be empty because this server was idle but on call.

We now discuss modifications to treat the bound η.

Each time a break is due, if the server is idle and the number of off-duty servers is less than η, then a break is announced and the server is not available to provide service for the duration θ. On the other hand, if the server is idle and the the number of off-duty servers equals η, then we give the server a high-priority designation and do not make the break announcement. Meanwhile, we keep track of the elapsed time since this high priority designation has been assigned.

At each customer departure epoch, if the queue is non-empty, then the server is assigned to the customer at the head of the queue. Hence, suppose that the queue is empty. If a high priority designation was given to that server and the number off-duty servers is less than η, then the break is announced and the server no longer provides service for the duration θ. Otherwise the server stays idle but on-call.

At each announced-break-end time, there is a newly-available server. We reset the server's age to zero. We assign this newly-available server to a customer if the queue is not empty. Otherwise, the newly-available server stays idle. At the meantime we look for other idle servers with a high-priority designation. If any, choose the one with the longest elapsed time since it received this high priority level and announce the break.

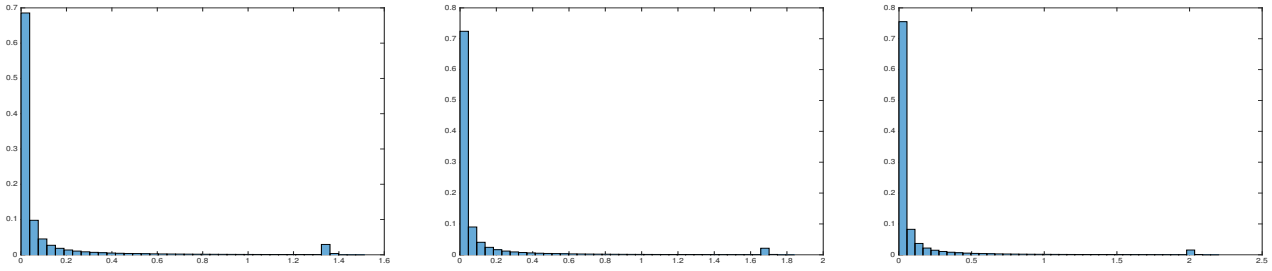
3 Additional Results for the D_1 Assignment Rule

In this section we provide additional simulation results for the D_1 assignment rule. In §3.1 we examine the impact of θ (target length-of-break) on system performance. We present more results for the $M/M/n$ queues in §3.2 and more results for the $M/H_2/n$ queues in §3.3.

3.1 Impact of θ on the D_1 Rule

From Lemma 2.1 in the paper, we know that the choice of the parameter θ directly influences the maximum possible rate at which breaks occur. Here we show via histograms that breaks occur less often as θ increases. Besides, as θ increases, the proportion of idleness on breaks also decreases.

The simulation results for the idle time distribution in the $M/M/n$ model with rule D_1 and model parameters $\mu = 1$, $\rho = 0.9$, $n = 100$ and three different values of θ: $\theta = k/3$ for $k = 4, 5, 6$ are displayed in Figure 1 and Table 1.



(a) $\theta = 4/3$ (b) $\theta = 5/3$ (c) $\theta = 2$

Figure 1: Histograms estimated from simulation of the idle-time distribution with rule $D_1(\theta)$ for three candidate targets θ

Panel (b) is for our base model with mean service times of 3 minutes, where the target duration was a 5-minute work break every one or two hours, so that $\theta = 5/3$. Evidently, the D_1 rule is able to create work breaks from idleness. Figure 1 shows that the D_1 rule creates a peak in the distribution at the target θ and the rest of the distribution concentrates near the origin, decaying very rapidly. Overall, we obtain a probability density function which is bimodal.

We elaborate in Table 1 by showing the 95% confidence intervals (based on the data collected from the simulation experiments) for the mean and standard deviation of the idle time V_n and the long-run proportion of idle times that are work breaks and $\pi_{\beta,I}$ to represent the long-run proportion of idle time that is made up of work break time.

Consistent with the conservation laws developed in the main paper, $E[V_n]$ approximately equals $1/9 = 0.1111$ for all θ, because the long-run proportion of idleness is solely determined by the traffic intensity ρ, independent of the server-assignment rule (provided that it is work-conserving). In addition, $std(V_n)$ increases as θ grows, but not significantly. Table 1 and Figure 1 also show that as θ increases from 1 to 2, the proportion of idle time occupied by breaks decreases slowly, changing from 0.486 to 0.294.

θ	$E[V_n]$	$std(V_n)$	$\pi_{\beta,I}$
6/6	$0.1112 \pm 5 \times 10^{-4}$	$0.2488 \pm 6 \times 10^{-4}$	0.486 ± 0.001
7/6	$0.1112 \pm 5 \times 10^{-4}$	$0.2653 \pm 7 \times 10^{-4}$	0.431 ± 0.001
8/6	$0.1113 \pm 4 \times 10^{-4}$	$0.2800 \pm 7 \times 10^{-4}$	0.398 ± 0.001
9/6	$0.1113 \pm 4 \times 10^{-4}$	$0.2928 \pm 7 \times 10^{-4}$	0.369 ± 0.001
10/6	$0.1108 \pm 5 \times 10^{-4}$	$0.3035 \pm 9 \times 10^{-4}$	0.340 ± 0.001
11/6	$0.1108 \pm 5 \times 10^{-4}$	$0.3138 \pm 1 \times 10^{-3}$	0.315 ± 0.002
12/6	$0.1112 \pm 5 \times 10^{-4}$	$0.3236 \pm 1 \times 10^{-3}$	0.294 ± 0.002

Table 1: Estimated performance measures of $D_1(\theta)$ as a function of θ

3.2 Impact of System Size on the D_1 Assignment Rule

The simulation results for the idle time distribution in the $M/M/n$ model with rule D_1 and model parameters $\mu = 1$, $\rho = 0.9$, $\theta = 5/3$ and 6 values of n ranging from 100 to 5000 are summarized in Table 2. See also the histograms and ECDFs in Figure 2 - 3 for rule D_1 as functions of n. Consistent with the fluid limit derived in the main paper, these histograms have a tendency to converge to the suggested form, an extremal two-point distribution with mass p on θ and mass $1 - p$ on 0. Here we recall that $p \equiv m/\theta = 0.0667$ with $m \equiv (1 - \rho)/\rho = 0.1111$. Moreover, from Table 2 we see that the probability values $P(V_n \geq \theta)$ have a tendency to converge to the limit $p = 0.0667$ as desired.

	$P(V_n \leq 0.001)$	$P(V_n \leq 0.01)$	$P(V_n \leq 0.1)$	$P(V_n \leq 1)$	$P(V_n \geq \theta)$
$n = 100$	0.2539 ± 0.0035	0.4629 ± 0.0027	0.8240 ± 0.0013	$0.9657 \pm 4 \times 10^{-4}$	$0.0223 \pm 4 \times 10^{-4}$
$n = 250$	0.1545 ± 0.0027	0.5270 ± 0.0017	$0.8498 \pm 7 \times 10^{-4}$	$0.9589 \pm 3 \times 10^{-4}$	$0.0317 \pm 3 \times 10^{-4}$
$n = 500$	0.1821 ± 0.0012	0.6228 ± 0.0010	$0.8717 \pm 8 \times 10^{-4}$	$0.9531 \pm 5 \times 10^{-4}$	$0.0405 \pm 5 \times 10^{-4}$
$n = 1000$	$0.2813 \pm 6 \times 10^{-4}$	$0.7093 \pm 7 \times 10^{-4}$	$0.8896 \pm 8 \times 10^{-4}$	$0.9474 \pm 7 \times 10^{-4}$	$0.0492 \pm 7 \times 10^{-4}$
$n = 2500$	$0.4618 \pm 5 \times 10^{-4}$	$0.7921 \pm 4 \times 10^{-4}$	$0.9074 \pm 5 \times 10^{-4}$	$0.9424 \pm 5 \times 10^{-4}$	$0.0564 \pm 5 \times 10^{-4}$
$n = 5000$	$0.5893 \pm 3 \times 10^{-4}$	$0.8333 \pm 3 \times 10^{-4}$	$0.9155 \pm 2 \times 10^{-4}$	$0.9395 \pm 2 \times 10^{-4}$	$0.0601 \pm 2 \times 10^{-4}$
$n = \infty$	0.9333	0.9333	0.9333	0.9333	0.0667

Table 2: Statistics for the idle-time distribution with rule D_1

Simulation outputs for the period between successive work breaks are reported in Table 3 and the histograms and ECDFs in Figure 4 - 5 as functions of n. In line with our asymptotic analysis, these histograms converge slowly to the desired form, i.e., a shifted exponential distribution. Specifically, the limit T can be expressed as $T \stackrel{\mathrm{d}}{=} x^* + \theta + M$ where M denotes an exponential r.v. with unit rate. Using the formulas derived in the main paper, we get $x^* = 1/p - 1 = 15 - 1 = 14$ and hence

$$\mathbb{E}[T] = x^* + 1 + \theta = 16.6667 \quad \text{and} \quad \sqrt{Var(T)} = \sqrt{Var(M)} = 1.$$

The first part of Table 3 shows strong evidence that both $\mathbb{E}[T_n]$ an $Var(T_n)$ converge to the correct limit as $s \to \infty$.

From the MHHT M/M fluid model with rule D_1, we expect that the age A_B of a busy server to follow a mixture distribution consisting of $U[0, \tau^*]$ and $\tau^* + M$ glued together on each side of the threshold $x^* = 14$ and the age A_I of an idle server to follow a mixture distribution consisting of a truncated-exponential and an exponential distribution spliced together back-to-back, as $n \to \infty$. The summary statistics for A_B and A_I are reported in Table 4. Figure 6

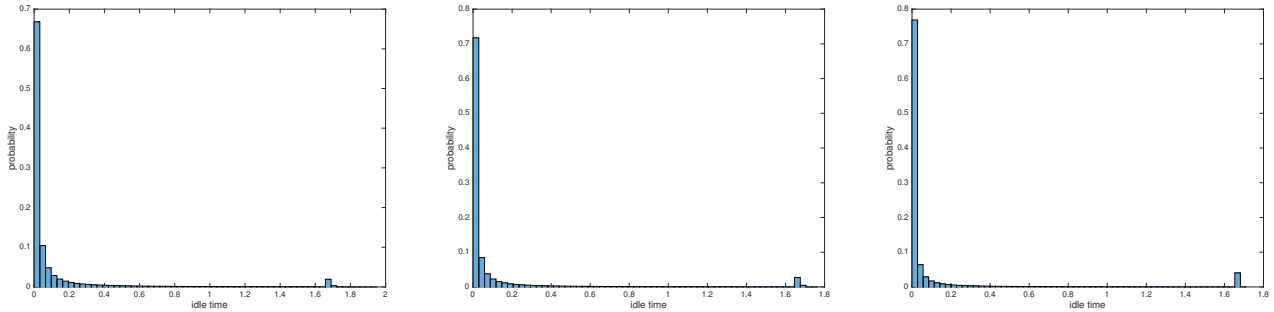
system	D_1		$SISF$	
size	$E[T_n]$	$std(T_n)$	$E[T_n]$	$std(T_n)$
$n = 100$	48.06 ± 0.79	18.73 ± 0.41	37.85 ± 0.49	36.68 ± 0.52
$n = 250$	33.45 ± 0.33	9.47 ± 0.35	28.62 ± 0.21	27.01 ± 0.28
$n = 500$	25.79 ± 0.34	5.66 ± 0.21	23.38 ± 0.20	21.65 ± 0.17
$n = 1000$	20.84 ± 0.30	3.06 ± 0.12	20.28 ± 0.16	18.54 ± 0.16
$n = 2500$	17.99 ± 0.14	1.75 ± 0.06	18.18 ± 0.09	16.46 ± 0.07
$n = 5000$	16.75 ± 0.07	1.38 ± 0.03	17.28 ± 0.05	15.59 ± 0.06
$n = \infty$	16.67	1.00	16.67	15.00

Table 3: Statistics for T_n with rule D_1

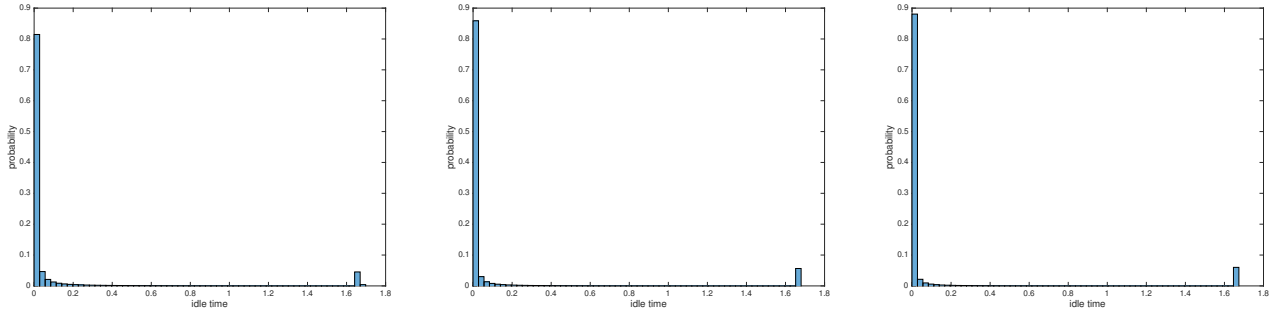
- 7 together with Table 4 strongly support the heavy-traffic fluid limits for A_B and A_I. See also the histograms as shown in Figure 6 - 7.

	Busy		Idle	
	$E[A_B]$	$std(A_B)$	$E[A_I]$	$std(A_I)$
$n = 100$	26.510 ± 0.055	19.146 ± 0.076	41.725 ± 0.073	19.725 ± 0.088
$n = 250$	17.602 ± 0.023	11.788 ± 0.045	31.131 ± 0.038	10.796 ± 0.041
$n = 500$	13.178 ± 0.016	8.395 ± 0.037	24.858 ± 0.022	6.565 ± 0.028
$n = 1000$	10.518 ± 0.010	6.380 ± 0.021	20.865 ± 0.014	3.828 ± 0.019
$n = 2500$	9.012 ± 0.006	5.349 ± 0.018	18.408 ± 0.009	2.406 ± 0.012
$n = 5000$	8.399 ± 0.004	4.935 ± 0.011	17.378 ± 0.005	1.797 ± 0.009

Table 4: Statistics for A_B and A_I ($M/M/n$ model)

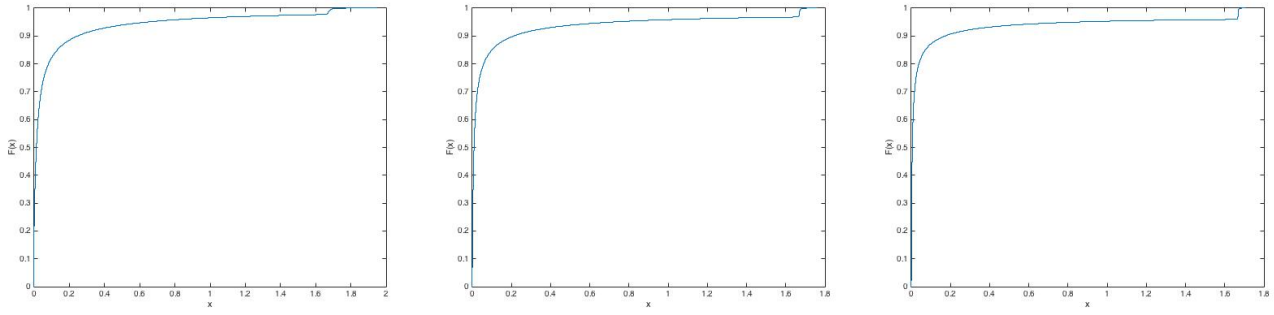


(a) $n = 100$ (b) $n = 250$ (c) $n = 500$

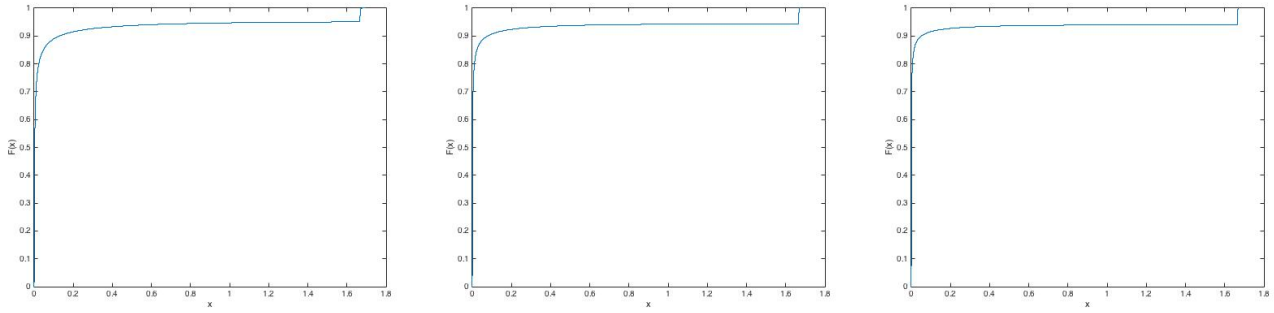


(d) $n = 1000$ (e) $n = 2500$ (f) $n = 5000$

Figure 2: Histogram of idle periods estimated from computer simulation with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$

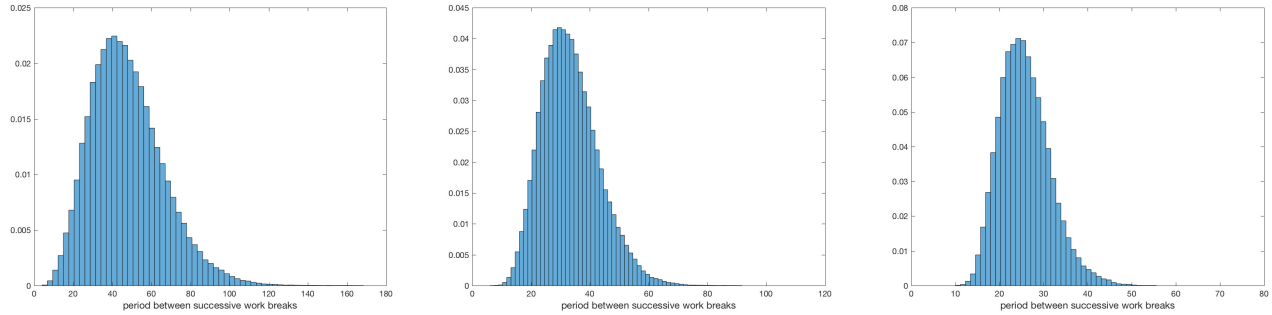


(a) $n = 100$ (b) $n = 250$ (c) $n = 500$

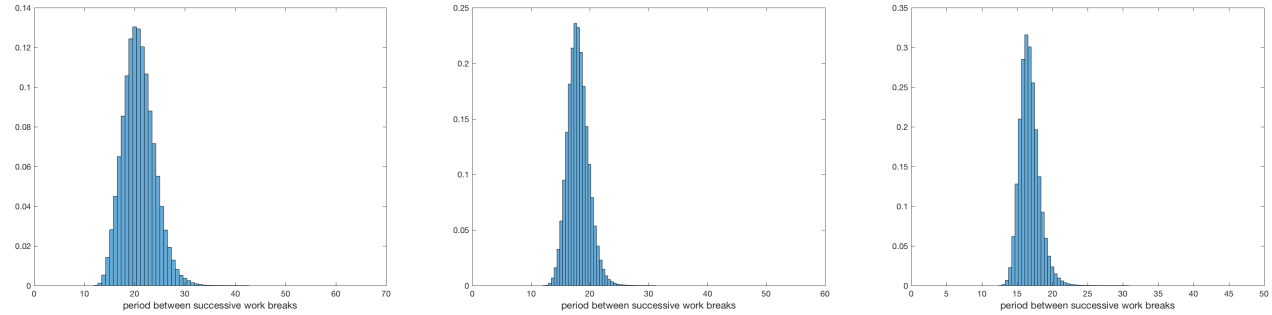


(d) $n = 1000$ (e) $n = 2500$ (f) $n = 5000$

Figure 3: ECDF of idle periods estimated from computer simulation with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$

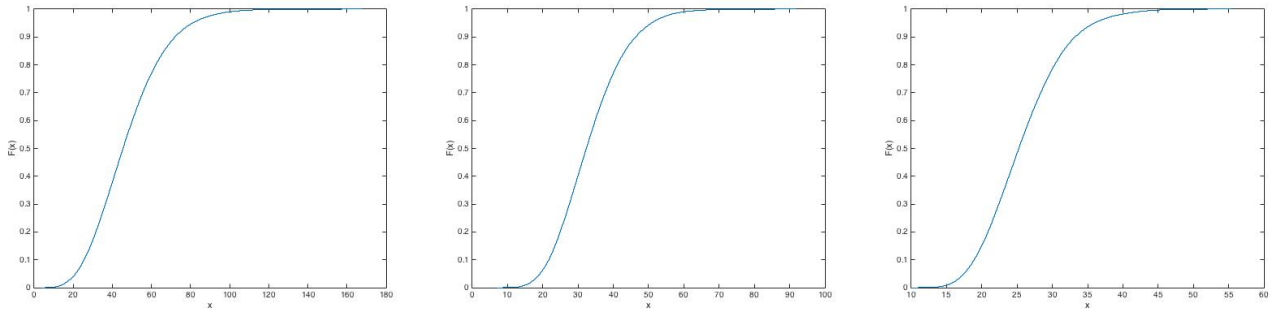


(a) $n = 100$ (b) $n = 250$ (c) $n = 500$

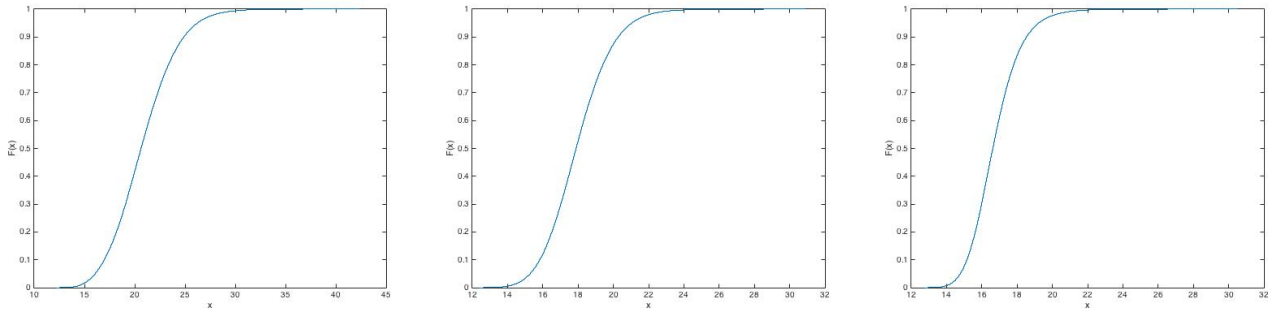


(d) $n = 1000$ (e) $n = 2500$ (f) $n = 5000$

Figure 4: Histogram of periods between successive breaks estimated from computer simulation with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$

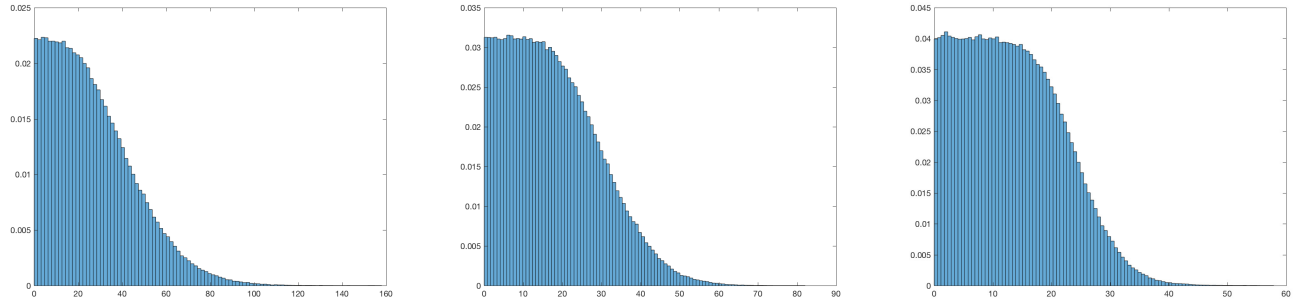


(a) $n = 100$ (b) $n = 250$ (c) $n = 500$

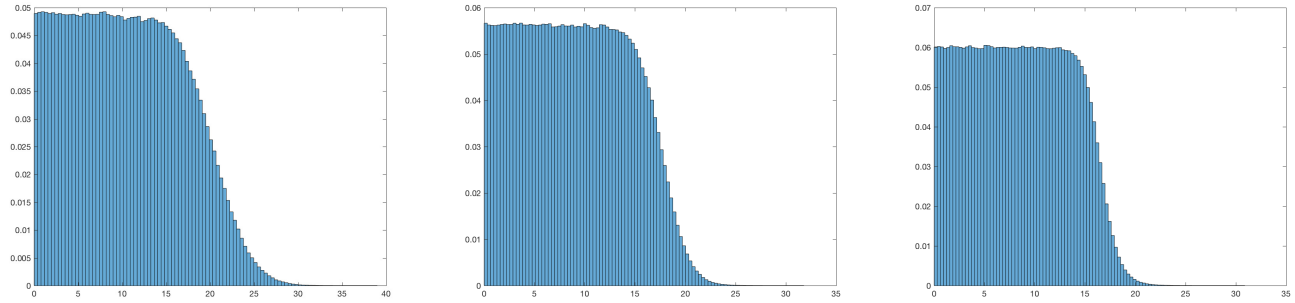


(d) $n = 1000$ (e) $n = 2500$ (f) $n = 5000$

Figure 5: ECDF of periods between successive breaks estimated from computer simulation with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$

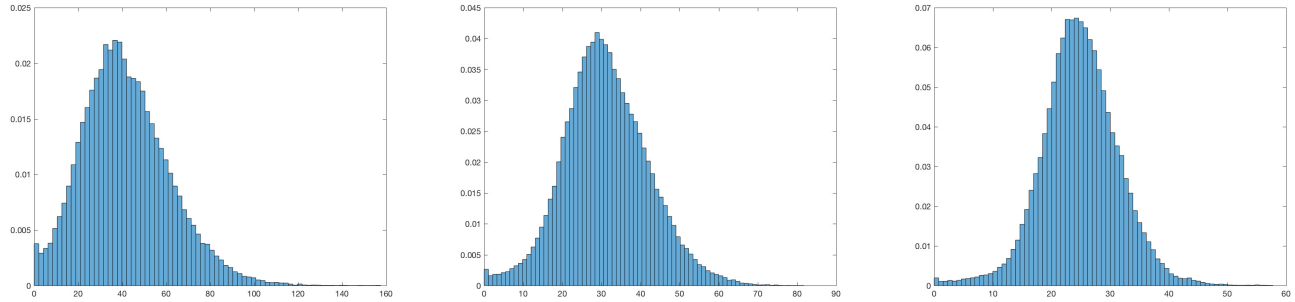


(a) $n = 100$ (b) $n = 250$ (c) $n = 500$

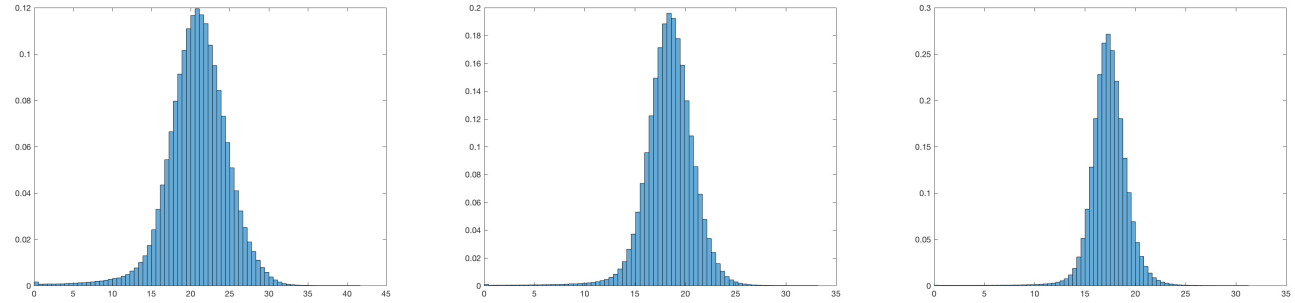


(d) $n = 1000$ (e) $n = 2500$ (f) $n = 5000$

Figure 6: Histogram of age of a busy server estimated from computer simulation for $M/M/n$ model with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$



(a) $n = 100$ (b) $n = 250$ (c) $n = 500$)



(d) $n = 1000$ (e) $n = 2500$ (f) $n = 5000$

Figure 7: Histogram of age of an idle server estimated from computer simulation for $M/M/n$ model with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$

3.3 Impact of Non-Markovian Service-Times

In Section 4.5 of the main paper, we briefly described how non-exponential service time distribution affects the period between breaks, T_n. The present section expands the discussion and servers two purposes: (a) validate the fluid limit of time between successive breaks, i.e.,

$$T \stackrel{\mathrm{d}}{=} x^* + R(x^*) + \theta = N(x^*) + 1 + \theta, \qquad \mathbb{E}[T] = m(x^*) + 1 + \theta = 1/\beta, \quad \text{and} \quad Var(T) = Var(R(x^*)),$$

where $N(\cdot), m(\cdot), R(\cdot), x^*$ and β are given in the paper when service times do not follow an exponential distribution; (b) expose the impact of non-exponential service times on the performance of rule D_1.

Here we assume the service time S to follow a hyper-exponential distribution or a mixture of exponential distributions; i.e., with probability π_j, the random variable S will take on the form of the exponential distribution with rate parameter μ_j. A hyper-exponential r.v. does not enjoy memoryless property and possesses higher variability (see e.g., Whitt (1982)). Besides, it is particularly convenient to work because the expression of all moments is very explicit. Indeed, the moment-generating function

$$E[e^{tS}] = \int_0^\infty f_S(x)\mathrm{d}x = \sum_{j=1}^K \pi_j \int_0^\infty e^{tx}\mu_j e^{-\mu_j x}\mathrm{d}x = \sum_{j=1}^K \pi_j\mu_j/(\mu_j - t) \tag{3.1}$$

from which we can easily compute the first three moments

$$E[S] = \sum_{j\le K} \pi_j/\mu_j, \qquad E[S^2] = \sum_{j\le K} 2\pi_j/\mu_j^2 \quad \text{and} \quad E[S^3] = \sum_{j\le K} 6\pi_j/\mu_j^3.$$

Let S^e be its associated stationary-excess distribution. From the simple relationship between S and S^e, we have

$$E[S^e] = \frac{E[S^2]}{2E[S]} = \frac{\sum_{j\le K}\pi_j/\mu_j^2}{\sum_{j\le K}\pi_j/\mu_j} \quad \text{and} \quad E[(S^e)^2] = \frac{E[S^3]}{3E[S]} = \frac{\sum_{j\le K} 2\pi_j/\mu_j^3}{\sum_{j\le K}\pi_j/\mu_j}.$$

As a result,

$$Var[S^e] = E[(S^e)^2] - (E[S^e])^2 = \frac{\sum_{j\le K} 2\pi_j/\mu_j^3}{\sum_{j\le K}\pi_j/\mu_j} - \left(\frac{\sum_{j\le K}\pi_j/\mu_j^2}{\sum_{j\le K}\pi_j/\mu_j}\right)^2$$

For our purpose, we'd like to construct a r.v. with desired mean 1 and coefficient of variation $c^2 = 4$. To match the first two moments, it suffices to consider a 2-phase hyper-exponential distribution. In particular we use $S = H_2$ with balanced means (see, e.g., Whitt (1982)). Applying the formulas in Whitt (1982), we get

$$\pi_1 = \left(1 + \sqrt{(c^2-1)/(c^2+1)}\right)/2 = 0.8873, \quad \mu_1 = 2\pi_1 = 1.7746,$$
$$\pi_2 = \left(1 - \sqrt{(c^2-1)/(c^2+1)}\right)/2 = 0.1127, \quad \mu_2 = 2\pi_2 = 0.2254.$$

With these parameters, we can easily compute the mean and variance of the stationary-excess distribution H_2^e. We expect the variance of time between successive idle periods to be close to that of H_2^e for s large enough. Using the expression as shown above, we obtain the first two moments of H_2^e:

$$E[H_2^e] = \pi_1/\mu_1^2 + \pi_2/\mu_2^2 = 2.5 \quad \text{and} \quad E[(H_2^e)^2] = 2\pi_1/\mu_1^3 + 2\pi_2/\mu_2^3 = 20$$

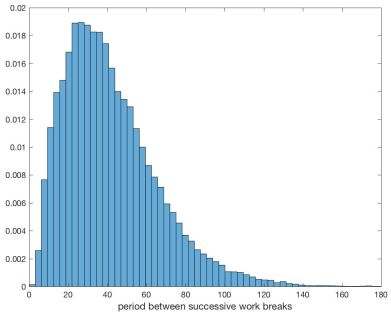
from which it follows

$$std(H_2^e) \equiv \sqrt{Var(H_2^e)} = \sqrt{20^2 - 2.5\times 2.5} = 3.7081.$$

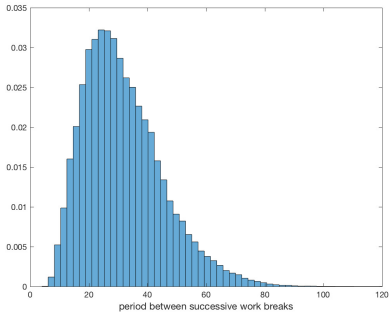
We elaborate in Table 5 by showing the 95% confidence intervals based on the data collected from the simulation experiments or the mean and the standard deviation of the interval between successive breaks, which we denoted by T_n. Consistent with the fluid limit with general service time distribution, the mean $\mathbb{E}[T_n]$ decreases as n grows and converges from above to the limit $\mathbb{E}[T] = 16.6667$; the standard deviation $std(T_n)$ also decreases in n and has a tendency to converge to the correct limit, e.g., $std(T_n) = 3.876$ for $s = 5000$, very close to the theoretic value 3.7081. As a supplement, Figure 8 displays the histograms of T_n estimated from computer simulation as a function in n. Consistent with the fluid limit, the distribution of T_n converges to the suggested form, i.e., a shifted excess-lifetime distribution $x^* + R(x^*) + \theta$.

	$E[A_B]$	$std(A_B)$	$E[A_I]$	$std(A_I)$	$E[T_n]$	$std(T_n)$
$n = 100$	27.145 ± 0.098	22.059 ± 0.102	37.622 ± 0.106	23.851 ± 0.115	41.663 ± 0.126	23.7531 ± 0.131
$n = 250$	18.277 ± 0.085	13.584 ± 0.092	29.473 ± 0.089	13.922 ± 0.079	31.748 ± 0.095	13.473 ± 0.104
$n = 500$	13.748 ± 0.082	9.654 ± 0.089	24.058 ± 0.084	9.049 ± 0.077	25.102 ± 0.087	8.587 ± 0.098
$n = 1000$	10.813 ± 0.062	7.249 ± 0.071	20.031 ± 0.075	5.883 ± 0.058	20.495 ± 0.047	5.568 ± 0.072
$n = 2500$	9.594 ± 0.043	6.288 ± 0.051	18.513 ± 0.055	4.439 ± 0.039	18.407 ± 0.034	4.228 ± 0.053
$n = 5000$	8.765 ± 0.022	5.789 ± 0.030	17.017 ± 0.028	4.150 ± 0.025	16.725 ± 0.024	3.876 ± 0.030

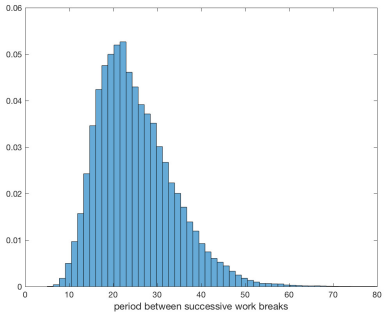
Table 5: Statistics for A_B, A_I and T_n for $M/H_2/n$ model with rule D_1, $\rho = 0.9$ and $\theta = 5/3$



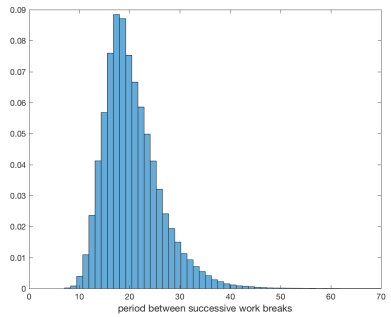
(a) $n = 100$



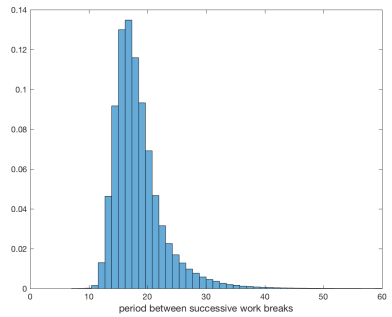
(b) $n = 250$



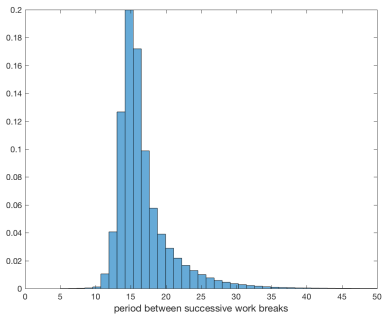
(c) $n = 500$



(d) $n = 1000$

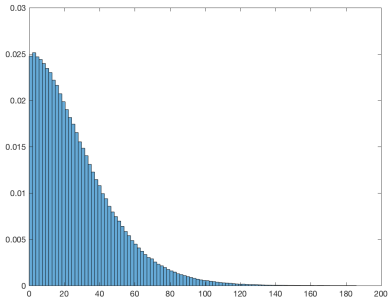


(e) $n = 2500$

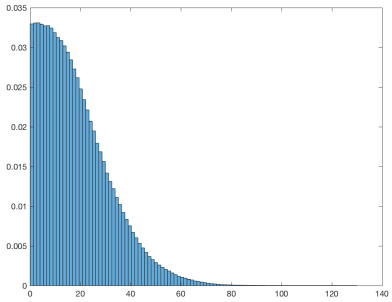


(f) $n = 5000$

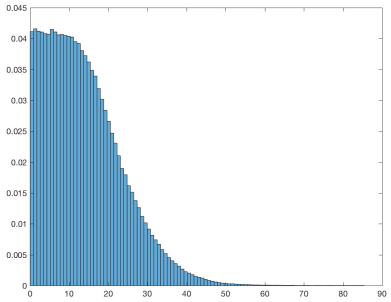
Figure 8: Histogram of T_n for $M/H_2/n$ model with rule D_1, $\rho = 0.9$ and $\theta = 5/3$



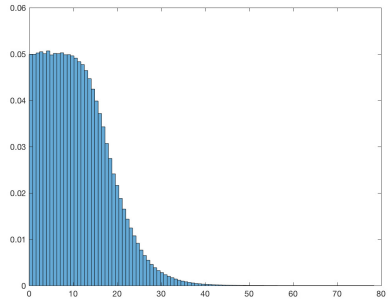
(a) $n = 100$



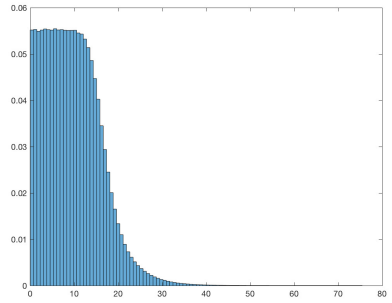
(b) $n = 250$



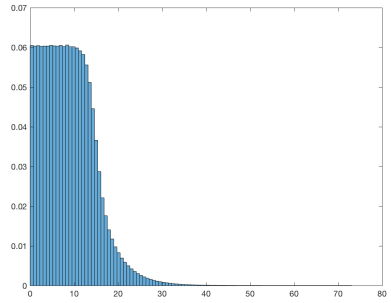
(c) $n = 500$



(d) $n = 1000$

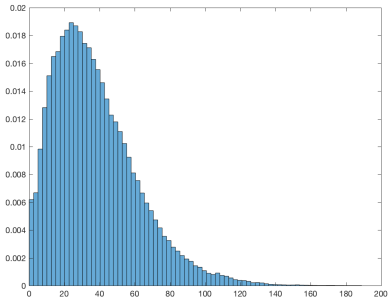


(e) $n = 2500$

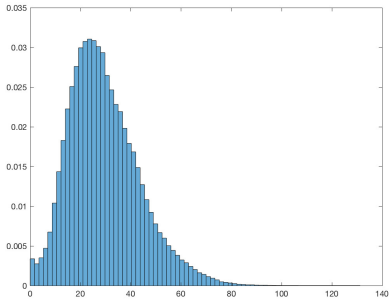


(f) $n = 5000$

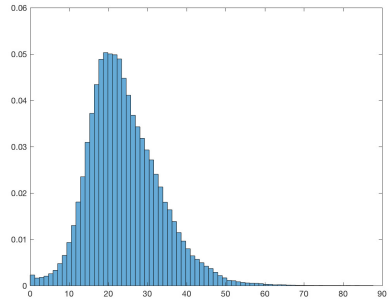
Figure 9: Histogram of age of a busy servers estimated from computer simulation for $M/H_2/n$ model with rule D_1, $\rho = 0.9$ and $\theta = 5/3$



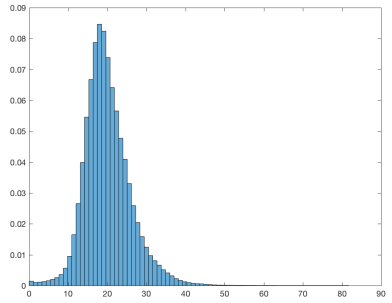
(a) $n = 100$



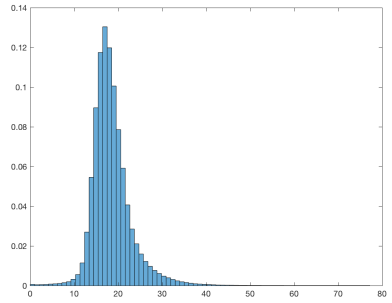
(b) $n = 250$



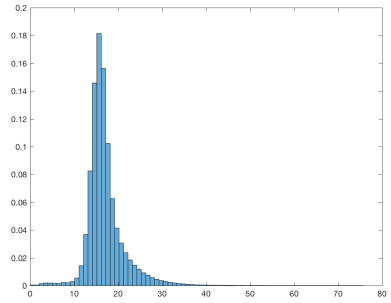
(c) $n = 500$



(d) $n = 1000$



(e) $n = 2500$



(f) $n = 5000$

Figure 10: Histogram of age of an idle server estimated from computer simulation for $M/H_2/n$ model with rule D_1 for $\rho = 0.9$ and $\theta = 5/3$

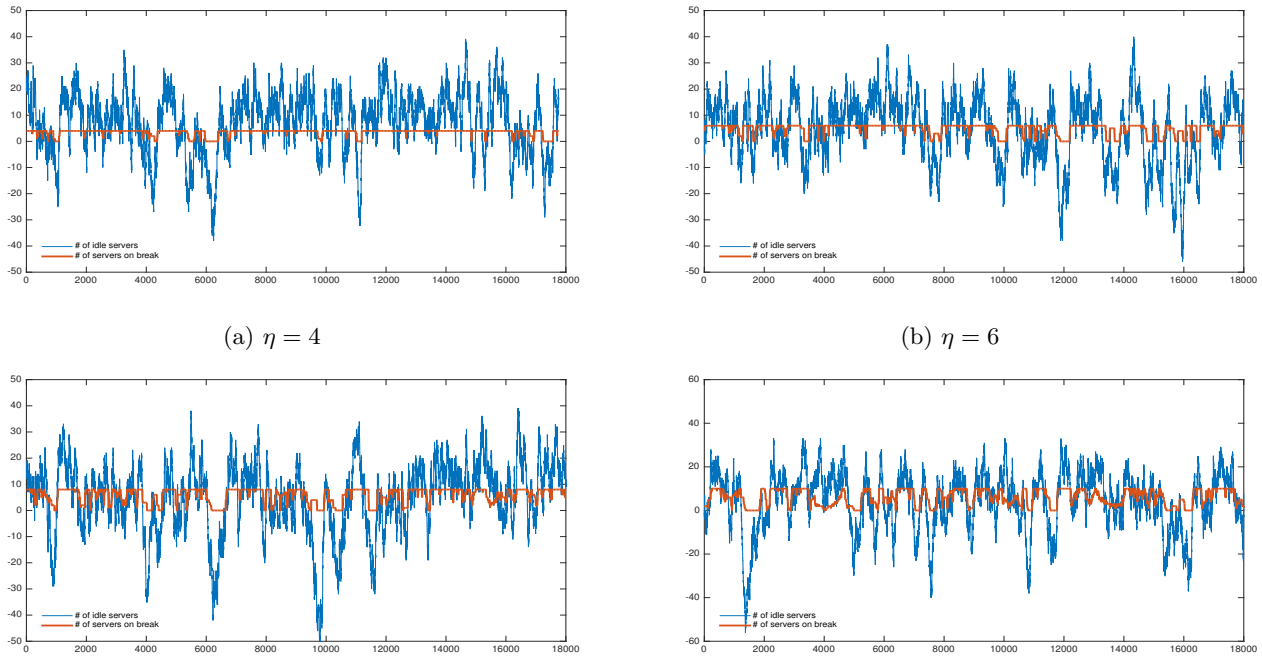
4 Additional Results on the D_2 Assignment Rule

The present section is to supplement Section 5 of the main paper. In §4.1 we examine the impact of the threshold parameter η on the system performance via appropriate sample paths. We present the simulation results for a large $M/M/n$ queue in §4.3.

4.1 Impact of the Parameter η

In Section 5.3 of the paper, we exposed the tradeoff in the choice of the parameter η through tables. In this section we show how the impact can be visualized through appropriate sample paths.

For greater insight, let $X(t)$ the number of customers in system at time t and $I_d(t) \equiv n - X(t)$ the number of idle servers at time t, allowing it to be negative as well as positive. Thus $-I_d(t) = Q(t)$, the queue length, when $I_d(t) < 0$, and $I(t) = I_d(t)^+$. We let $S_b(t)$ be the number of servers on break at time t. Figure 11 displays sample paths of the number of servers on break, $S_b(t)$, and the number of idle servers, $I_d(t)$, for the base $M/M/n$ model with $n = 100$, $\rho = 0.9$, and four different values of η when $\tau = 20$ and $\theta = 5/3$. Panel (c) with $\eta = 8$ shows a severe performance degradation for customers because we often observe a big downward spike in $I_d(t)$, which suggests a buildup of large queue.



(a) $\eta = 4$ (b) $\eta = 6$

(c) $\eta = 8$ (d) $\eta = 10$

Figure 11: Sample paths of the number of servers on break, $S_b(t)$ (red), and the number of idle servers, $I_d(t) \equiv s - N(t)$ (blue), with rule D_2 as a function of η for $\theta = 5/3$ and $\tau = 20$

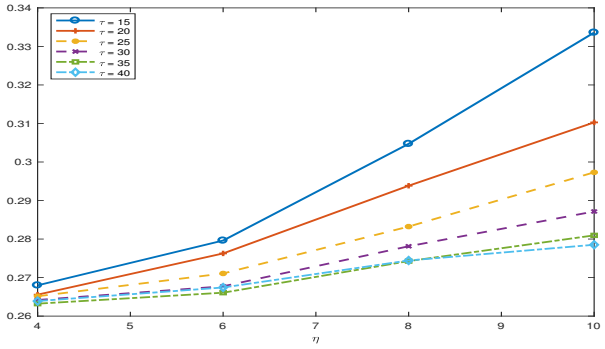
4.2 A Small System

In the main paper, we formulate an optimization to choose the parameters τ and η. In particular, we suggest performing a simple optimization with a cost function that is a convex weighted sum of $1 - p_A$ and $p_D - p_D^*$, i.e.,

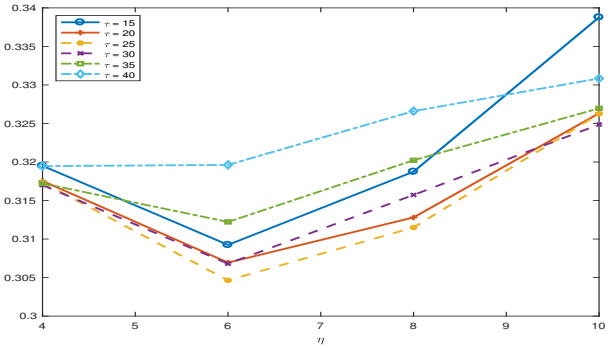
$$C \equiv C(p_A, p_D) \equiv w(1 - p_A) + (1 - w)(p_D - p_D^*), \quad 0 \le w \le 1, \tag{4.1}$$

where p_D^* is the LISF value, which is 0.223 for $n = 100$ and 0.001 for $n = 1000$ and the weight w reflects the relative cost we wish to attribute to p_A versus p_D. In the main paper, we displayed and discussed the simulation results for the base model with $n = 100$. $\rho = 0.9$ and weight $w = 0.5$. The present section provides addition numerical results with different weight parameters.

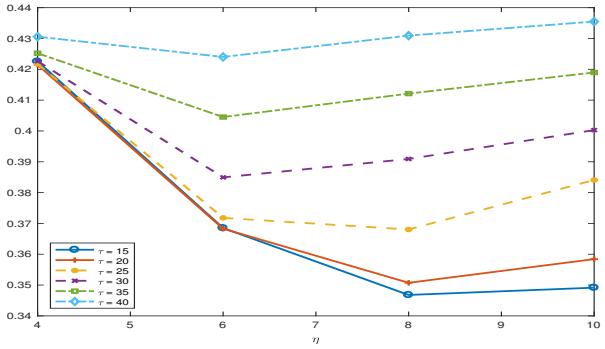
Figure 12 shows the cost C as a function of τ and η for $n = 100$, $\theta = 5/3$ and four weights $w = 0.3, 0.4, 0.6, 0.7$. Panel 12a shows that for $w = 0.3$ the optimal (τ^*, η^*) is attained at $(\tau = 40, \eta = 4)$. Panel 12b shows that for $w = 0.4$ $(\tau = 25, \eta = 6)$ is the optima. Panel 12c suggests for $w = 0.6$ that $(\tau = 15, \eta = 8)$ is the optimal combination while 12d show that for $w = 0.7$, the choice $(\tau = 15, \eta = 10)$ is favorable.



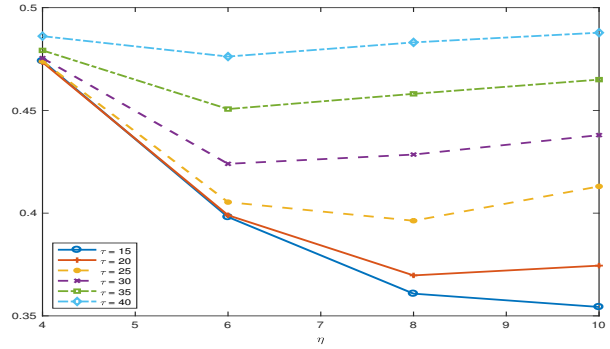
(a) $w = 0.3$



(b) $w = 0.4$



(c) $w = 0.6$



(d) $w = 0.7$

Figure 12: Cost function for $D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $n = 100$, $\theta = 5/3$ and $w = 0.3, 0.4, 0.6, 0.7$

4.3 A Large System

We now amplify Remark 5.1 in the main paper by simulating a large $M/M/n$ queue with $n = 1000$ and $\rho = 0.9$.

Table 6 and 7 display two basic performance measures, namely the delay probability p_D and the proportion of idle on work breaks p_A as a function of τ and η. Estimates of the 95% confidence intervals are displayed in the tables.

As the system size grows larger, both types of costs are significant reduced, and yet there is tradeoffs in the choice of (τ, η). Again both p_D and p_A are monotone in τ, but p_A is not monotone in the bound η. For τ ranging from 15 to 30, the largest value of p_A is attained at $\eta = 90, 80, 70$ and 60 respectively. These values are highlighted in Table 7.

	$\eta = 4$	$\eta = 6$	$\eta = 8$	$\eta = 10$
τ	p_D	p_D	p_D	p_D
$\tau = 15$	0.0976 ± 0.0024	0.1630 ± 0.0029	0.2427 ± 0.0052	0.3203 ± 0.0030
$\tau = 20$	0.0892 ± 0.0019	0.1364 ± 0.0019	0.1487 ± 0.0021	0.1611 ± 0.0033
$\tau = 25$	0.0699 ± 0.0016	0.0742 ± 0.0015	0.0808 ± 0.0023	0.0854 ± 0.0023
$\tau = 30$	0.0422 ± 0.0010	0.0529 ± 0.0014	0.0612 ± 0.0016	0.0720 ± 0.0011

Table 6: 95% confidence intervals of delay probability for rule $D_2(\theta, \tau, \eta)$ as a function of and τ and η for $n = 1000$ and $\theta = 5/3$.

	$\eta = 4$	$\eta = 6$	$\eta = 8$	$\eta = 10$
τ	p_A	p_A	p_A	p_A
$\tau = 15$	0.7061 ± 0.0013	0.7700 ± 0.0011	0.8114 ± 0.0009	$\mathbf{0.8231 \pm 0.0018}$
$\tau = 20$	0.7033 ± 0.0012	0.7631 ± 0.0012	$\mathbf{0.7808 \pm 0.0006}$	0.7773 ± 0.0011
$\tau = 25$	0.6851 ± 0.0015	$\mathbf{0.6938 \pm 0.0014}$	0.6860 ± 0.0024	0.6780 ± 0.0024
$\tau = 30$	$\mathbf{0.5850 \pm 0.0017}$	0.5593 ± 0.0026	0.5422 ± 0.0028	0.5368 ± 0.0021

Table 7: 95% confidence intervals for proportion of idle time spent on announced work breaks for rule $D_2(\theta, \tau, \eta)$ as a function of and τ and η for $n = 1000$ and $\theta = 5/3$.

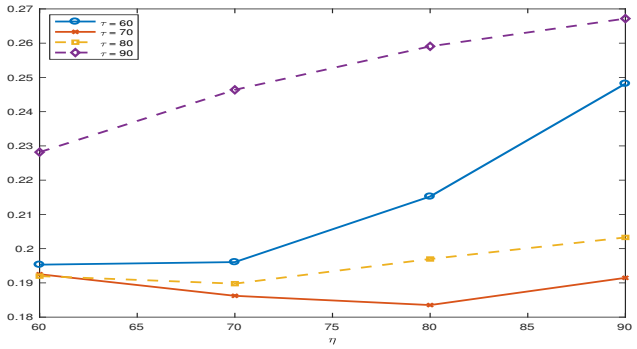
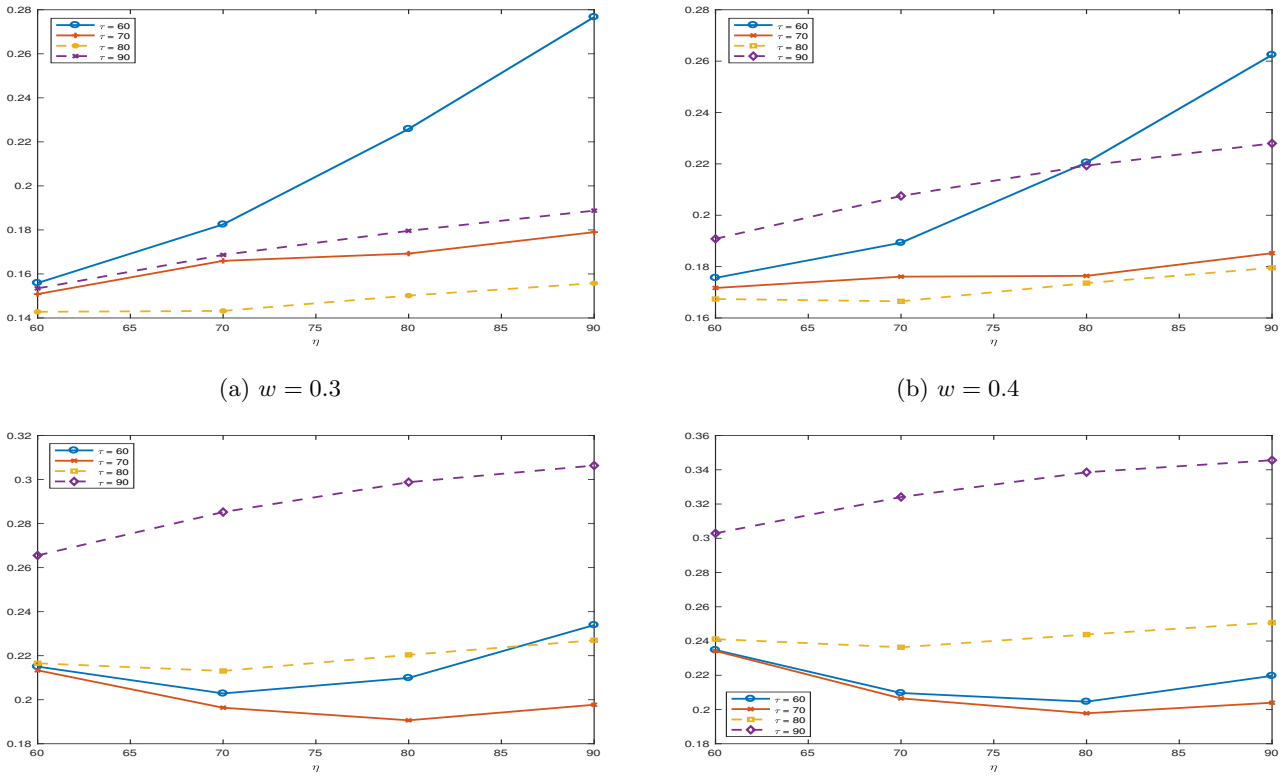


Figure 13: Cost function for $D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $n = 1000$, $\theta = 5/3$ and $w = 0.5$

Solve the optimization problem (4.1) gives Figure 14 shows the cost C as a function of τ and η for $n = 1000$, $\theta = 5/3$ and four weights $w = 0.3, 0.4, 0.6, 0.7$. Panel 14b shows that for $w = 0.4$ the combination $(\tau = 80, \eta = 70)$ is optimal whereas Panel 14c shows that for $w = 0.6$, the choice $(\tau = 70, \eta = 80)$ is more desirable.



(a) $w = 0.3$ (b) $w = 0.4$

(c) $w = 0.6$ (d) $w = 0.7$

Figure 14: Cost function for $D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $n = 1000$, $\theta = 5/3$ and $w = 0.3, 0.4, 0.6, 0.7$

4.4 Comparison with the Standard $M/M/(n-b)$ Model

An alternative way to obtain work breaks is to place a constant number of servers on break. If we put b servers on break at all times, then we obtain an $M/M/(n-b)$ model with the customary LISF server assignment rule. It is useful to compare $D_2(\theta, \tau, \eta)$ to an $M/M/(s-b)$ LISF model by considering a range of b from $\lfloor E[S_b] \rfloor$, the greatest integer less than or equal to $E[S_b]$, to η. Table 8 displays $E[S_b]$ as a function of η estimated from computation simulations.

We find that $D_2(\theta, \tau, \eta)$ outperforms $M/M/(s-b)$ LISF for $\lfloor E[S_b] \rfloor \leq b \leq \eta$. For example, when $\theta = 5/3$, $\tau = 20$ and $\eta = 8$, Table 8 shows that $E[S_b] = 5.610$. Table 6 of the main paper shows that $p_D = 0.460$ which is less than the values for $b = 5$ in Table 9, and far less than the value for $b = \eta = 8$.

η	1	2	3	4	5	6	7	8	9
$E[S_b]$	0.908	1.789	2.630	3.422	4.166	4.834	5.358	5.610	5.708

Table 8: Estimated mean of the number of servers on announced breaks for the D_2 rule as a function of η for $\theta = 5/3, \tau = 20$

b	0	1	2	3	4	5	6	7	8
p_D	0.216	0.257	0.304	0.358	0.420	0.488	0.564	0.648	0.737
$E[Q]$	1.90	2.51	3.33	4.45	6.00	8.23	11.54	16.68	25.19
$std(Q)$	5.62	6.69	8.01	9.65	11.75	14.49	18.20	23.48	31.00

Table 9: Performance measures for the standard $M/M/(100-b)$ queue with $\rho = 0.9$

5 The $LISF - D_2$ Assignment Rule

In Remark 5.2 of the paper we mentioned an alternative to the D_2 assignment rule that is easier to implement, and has similar performance. We now amplify this remark and describe in detail the alternative rule which we refer to as the $LISF - D_2$ assignment rule. We show that its performance is similar to the D_2 rule elaborated in the main paper.

Under the LISF rule there is a *FIFO queue for assignment*, i.e., whoever becomes idle first gets assigned first. Now we maintain two FIFO queues for assignment, a high priority queue (HPQ) and a low priority queue (LPQ). The rule stipulates that servers join the back of the HPQ once finishing a break.

5.1 Implementation of the $LISF - D_2$ Rule

There four types of events: customer arrival, customer departure (service completion), due for a break and work-break completion. We first explain how to treat the control parameter τ with $\eta = \infty$, so it plays no role. Afterwards, we discuss the modifications to include η.

At each arrival epoch, we look for idle servers in the HPQ. If any, assign the server at the head of the HPQ. Otherwise we look for idle servers in the LPQ. If any, assign the server at the head of the LPQ. For the selected idle server, the algorithm generates a service requirement S and resets its service completion time to $t + S$. Then we find the minimum service-completion time among all busy servers and update the departure time accordingly. If there are no idle servers, the arriving customer waits in queue.

At each departure epoch, we look for customers in queue. If there is customer waiting, assign the server to the head-of-line customer. Otherwise the server either becomes idle or starts a break depending on whether or not a high priority designation was given. If a high priority designation was given, the break is announced and the server is not available to provide service for the duration θ after that time. Otherwise it joins the back of the LPQ.

At each break due time (when a server's age reaches τ), if the server is busy, then we give the server a high priority designation indicating that that its next idle period will be replaced by an announced break. If the server is idle, then the server starts a break and goes off duty for the duration θ.

At each break-end time, we first reset the server's age to zero. We assign to it a customer if there are customers in queue. Otherwise the server joins the back of HPQ. This prevents work break from being much greater than θ since we always make assignment from the HPQ first.

We now discuss modifications to treat the bound η.

Each time a break is due, if the server is busy, we assign it a high priority designation. Meanwhile, we keep track of the elapsed time since this high priority designation has been assigned. If the server is idle and the number of

off-duty servers is less than η, then a break is announced and the server is not available to provide service for the duration θ. On the other hand, if the server is idle and the number of off-duty servers equals η, then we give the server a high-priority designation and do not make break announcement; and again keep track of the elapsed time since this high priority designation has been assigned..

At each departure epoch, if the queue is non-empty, then the server is assigned to the customer at the head of the queue. Hence suppose that the queue is empty. If a high priority designation was given *and* to the server and the number off-duty servers is less than η, the break is announced and the server no longer provides service for the duration θ. Otherwise the server joins the back of the LPQ.

At each break-end time, we first reset the server's age to zero. We assign to it a customer if there are customers in queue. Otherwise the server joins the back of HPQ. Meanwhile, we look for idle servers with high priority designation. If any, choose the one with the longest elapsed time since it receives this high priority level and announce the break.

5.2 A Small System

We now study the impact of the control parameter τ and η for $n = 100$ and $\theta = 5/3$ with the $LISF - D_2$ rule. Table 10 and 11 display two basic performance measures, namely the delay probability p_D and the proportion of idle on work breaks p_A as a function of τ and η. In particular, estimates of the 95% confidence intervals are shown in the tables.

We see that there is a strong tradeoff in the choice of η, for a given τ, between the effectiveness of the breaks for the servers and the performance experienced by customers. That tradeoff is dramatically in the two tables. For $\tau = 20$ and $\eta = 4$, there is moderate performance degradation for customers with a delay probability 0.3353, but the algorithm for work breaks is ineffective, e.g., only a third of the available idleness is turned into work breaks. On the other hand for $\tau = 20$ and $\eta = 10$, the algorithm is very effective in generating work breaks, i.e., more than half of the total idleness is turned into work breaks, but there is severe performance degradation for customers, e.g., p_D increases by 50% reaching 0.4841.

Both p_D and p_A are monotone in τ, but p_A is not monotone in the bound η. For $\tau = 4-6$, $7-8$ and $9-10$, the largest value of p_A is attained at $\eta = 10, 8$ and 6 respectively. These values are highlighted in Table 11.

	$\eta = 4$	$\eta = 6$	$\eta = 8$	$\eta = 10$
τ	p_D	p_D	p_D	p_D
$\tau = 15$	0.3363 ± 0.0023	0.4130 ± 0.0025	0.4873 ± 0.0023	0.5397 ± 0.0021
$\tau = 20$	0.3353 ± 0.0016	0.4065 ± 0.0020	0.4594 ± 0.0031	0.4841 ± 0.0026
$\tau = 25$	0.3325 ± 0.0020	0.3934 ± 0.0025	0.4198 ± 0.0022	0.4342 ± 0.0024
$\tau = 30$	0.3323 ± 0.0021	0.3728 ± 0.0023	0.3853 ± 0.0029	0.3979 ± 0.0030
$\tau = 35$	0.3243 ± 0.0021	0.3531 ± 0.0017	0.3583 ± 0.0016	0.3670 ± 0.0026
$\tau = 40$	0.3195 ± 0.0026	0.3314 ± 0.0026	0.3381 ± 0.0026	0.3446 ± 0.0020

Table 10: 95% confidence intervals for delay probability of $LISF - D_2(\theta, \tau, \eta)$ as a function of and τ and η for $n = 100$ and $\theta = 5/3$.

In order to choose the parameters τ and η, we again solve the optimization with the cost function as shown in

	$\eta = 4$	$\eta = 6$	$\eta = 8$	$\eta = 10$
τ	p_A	p_A	p_A	p_A
$\tau = 15$	$0.3343 \pm 7 \times 10^{-4}$	$0.4548 \pm 6 \times 10^{-4}$	$0.5357 \pm 7 \times 10^{-4}$	$\mathbf{0.5726 \pm 7 \times 10^{-4}}$
$\tau = 20$	$0.3323 \pm 7 \times 10^{-4}$	$0.4475 \pm 8 \times 10^{-4}$	$0.5081 \pm 9 \times 10^{-4}$	$\mathbf{0.5171 \pm 9 \times 10^{-4}}$
$\tau = 25$	$0.3300 \pm 7 \times 10^{-4}$	$0.4308 \pm 9 \times 10^{-4}$	$\mathbf{0.4558 \pm 9 \times 10^{-4}}$	$0.4527 \pm 9 \times 10^{-4}$
$\tau = 30$	$0.3270 \pm 4 \times 10^{-4}$	$0.3998 \pm 6 \times 10^{-4}$	$\mathbf{0.4035 \pm 7 \times 10^{-4}}$	$0.3998 \pm 9 \times 10^{-4}$
$\tau = 35$	$0.3205 \pm 6 \times 10^{-4}$	$\mathbf{0.3620 \pm 6 \times 10^{-4}}$	$0.3598 \pm 9 \times 10^{-4}$	$0.3542 \pm 8 \times 10^{-4}$
$\tau = 40$	$0.3097 \pm 7 \times 10^{-4}$	$\mathbf{0.3262 \pm 9 \times 10^{-4}}$	$0.3225 \pm 9 \times 10^{-4}$	$0.3186 \pm 9 \times 10^{-4}$

Table 11: 95% confidence intervals for proportion of idle time spent on announced work breaks for rule $LISF - D_2(\theta, \tau, \eta)$ as a function of and τ and η for $n = 100$ and $\theta = 5/3$.

(4.1) Figure 16 shows the cost C as a function of τ and η for $n = 100$, $\theta = 5/3$ and four weights $w = 0.3, 0.4, 0.6, 0.6$. Panel 16a shows that for $w = 0.3$ the optimal (τ^*, η^*) is attained at $(\tau = 40, \eta = 4)$. Panel 16b shows that for $w = 0.4$ both $(\tau = 25, \eta = 6)$ and $(\tau = 30, \eta = 6)$ are optimal. Panel 16c - 16d show that for $w \geq 0.6$, the choice $(\tau = 15, \eta = 10)$ is favorable.

Comparing Table 10 with Table 6 in the main paper, we see that the two approaches, i.e., the $LISF - D_2$ and the D_2 rule, are comparable in terms of performance degradation caused by enforced breaks; but putting Table 11 in contrast with Table 5 in the main paper, we observe that the D_2 rule consistently outperforms the $LISF - D_2$ rule by a small margin in terms of their effectiveness in generating announced breaks.

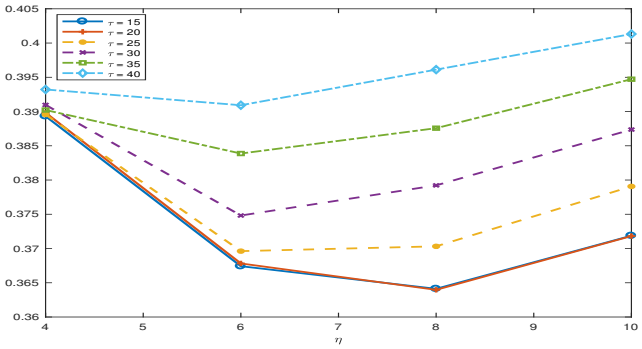


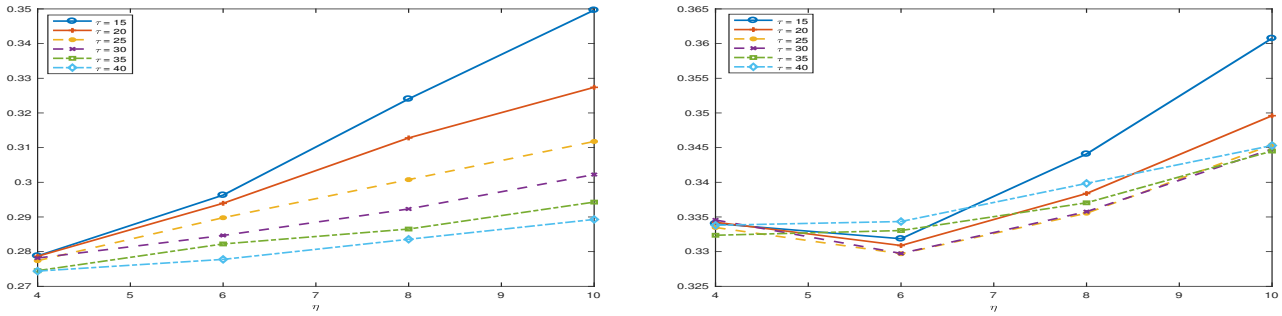
Figure 15: Cost function for $LISF - D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $n = 100$, $\theta = 5/3$ and $w = 0.5$

5.3 A Large System

Here we fix traffic intensity $\rho = 0.9$ and let system size n grows. Particularly we consider $n = 1000$ and hence $\lambda = 900$.

Table 12 and 13 display two basic performance measures, namely the delay probability p_D and the proportion of idle on work breaks p_A as a function of τ and η with the $LISF - D_2$ rule. Estimates of the 95% confidence intervals are displayed in the tables.

We see that with the system size being greater, both types of costs are significant reduced, and yet there is



(a) $w = 0.3$ (b) $w = 0.4$

(c) $w = 0.6$ (d) $w = 0.7$

Figure 16: Cost function for $LISF - D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $s = 100$, $\theta = 5/3$ and $w = 0.3, 0.4, 0.6, 0.7$

tradeoffs in the choice of (τ, η). When $\tau = 15$, for example, the choice of η exert a great influence on the effectiveness of the breaks for the servers and the performance experienced by customers.

Again both p_D and p_A are monotone in τ, but p_A is not monotone in the bound η. For τ ranging from 15 to 30, the largest value of p_A is attained at $\eta = 90, 80, 70$ and 60 respectively. These values are highlighted in Table 13.

	$\eta = 60$	$\eta = 70$	$\eta = 80$	$\eta = 90$
τ	p_D	p_D	p_D	p_D
$\tau = 15$	0.0946 ± 0.0022	0.1628 ± 0.0032	0.2471 ± 0.0035	0.3162 ± 0.0034
$\tau = 20$	0.0907 ± 0.0022	0.1348 ± 0.0023	0.1512 ± 0.0034	0.1624 ± 0.0031
$\tau = 25$	0.0706 ± 0.0018	0.0731 ± 0.0016	0.0765 ± 0.0018	0.0803 ± 0.0018
$\tau = 30$	0.0407 ± 0.0011	0.0430 ± 0.0017	0.0433 ± 0.0015	0.0435 ± 0.0014

Table 12: 95% confidence intervals for delay probability of $LISF - D_2(\theta, \tau, \eta)$ as a function of and τ and η for $n = 1000$ and $\theta = 5/3$.

To determine the optimal (τ^*, η^*), we again solve the optimization problem with a cost function as given in (4.1). Figure 18 shows the cost C as a function of τ and η for $n = 1000$, $\theta = 5/3$ and four weights $w = 0.3, 0.4, 0.6, 0.7$. Panel 18a and 18d correspond to two extreme cases which are not very interesting. Panel 18b shows that for $w = 0.4$ the combination $(\tau = 80, \eta = 70)$ is optimal. whereas Panel 18c shows that for $w = 0.6$, the choice $(\tau = 70, \eta = 80)$ is more desirable.

	$\eta = 60$	$\eta = 70$	$\eta = 80$	$\eta = 90$
τ	p_A	p_A	p_A	p_A
$\tau = 15$	0.5951 ± 0.0015	0.6766 ± 0.0016	0.7359 ± 0.0011	$\mathbf{0.7663 \pm 0.0015}$
$\tau = 20$	0.5940 ± 0.0013	0.6642 ± 0.0011	$\mathbf{0.6822 \pm 0.0010}$	0.6797 ± 0.0009
$\tau = 25$	0.5771 ± 0.0009	$\mathbf{0.5823 \pm 0.0009}$	0.5807 ± 0.0018	0.5800 ± 0.0016
$\tau = 30$	$\mathbf{0.5015 \pm 0.0014}$	0.5007 ± 0.0017	0.5001 ± 0.0014	0.4994 ± 0.0013

Table 13: 95% confidence intervals for proportion of idle time spent on announced work breaks for rule $LISF - D_2(\theta, \tau, \eta)$ as a function of and τ and η for $n = 1000$ and $\theta = 5/3$.

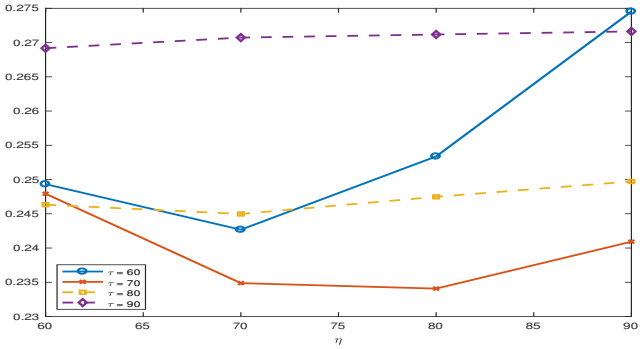
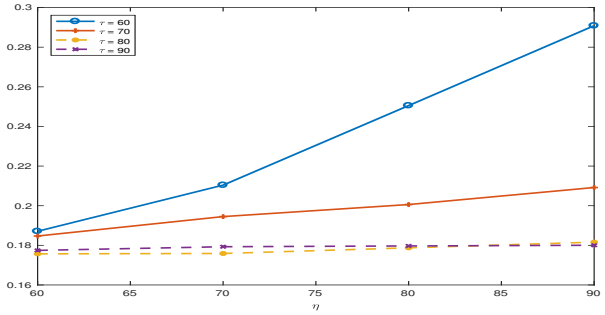
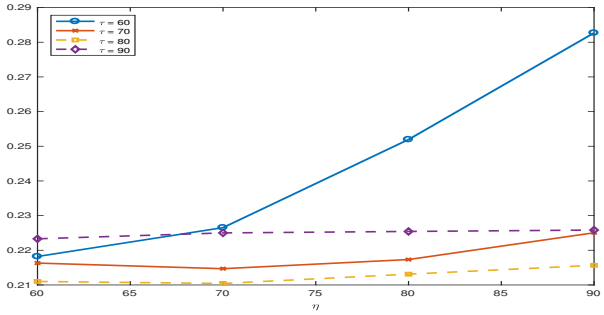


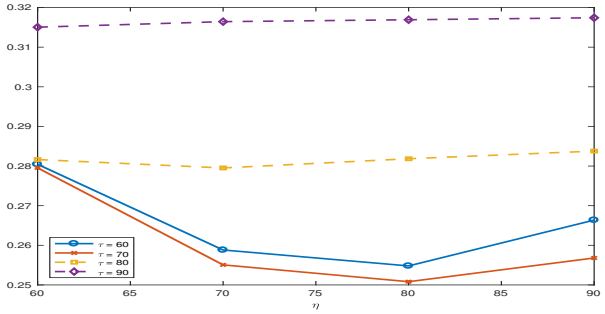
Figure 17: Cost function for $LISF - D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $n = 1000$, $\theta = 5/3$ and $w = 0.5$



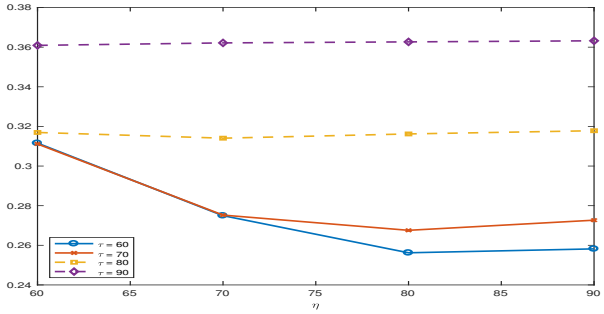
(a) $w = 0.3$



(b) $w = 0.4$



(c) $w = 0.6$



(d) $w = 0.7$

Figure 18: Cost function for $LISF - D_2(\theta, \tau, \eta)$ in (4.1) as a function of τ and η for $s = 1000$, $\theta = 5/3$ and $w = 0.3, 0.4, 0.6, 0.7$

6 On the Existence of a Critical Threshold for D_1

In this final section we supplement the discussion in §3.1 of the main paper on the existence of a critical threshold for the fluid model with the server-assignment rule D_1.

Let $I(x)$ denote the amount of idle server fluid with age at most x in steady state. Because the idle fluid that has been idle for the least time is assigned first, there exists $\tau \geq 0$ such that $I(x) > 0$ for all $x > \tau$ and $I(x) = 0$ for all $x < \tau$. For the busy server fluid, let $B(x, y)$ be the busy server fluid contend with age at most x and elapsed service time at most y (in steady state) and write

$$B(x,y) \equiv \int_0^x \int_0^y b(u,v)dvdu.$$

If $I(x) > 0$, then there is idle server fluid content with age less than or equal to x. With the D_1 rule, any busy server fluid with age greater than x will not be reassigned. Hence

$$\int_x^\infty \int_0^{u-\tau} b(u,v)dvdu = 0 \quad \text{for} \quad x < \tau \tag{6.1}$$

and

$$\frac{d}{dx}\left[\int_x^\infty \int_{u-\tau}^u b(u,v)dvdu\right] = -\int_x^\infty \int_{u-\tau}^u b(u,v)h(v)dvdu \quad \text{for} \quad x > \tau, \tag{6.2}$$

where $h(\cdot)$ denotes the hazard-rate function of the service-time distribution. In view of (6.1), the l.h.s. of (6.2) is simply $-b_a(x)$ where we have used $b_a(x)$ to represent the the density of the age, which is the marginal density of $b(x, y)$. For the r.h.s., we have

$$\begin{aligned}
\int_x^\infty \int_{u-\tau}^u b(u,v)h(v)dvdu &= \int_x^\infty \int_{u-\tau}^u b(u,v)\frac{f(v)}{F^c(v)}dvdu \\
&= \int_x^\infty \int_{u-\tau}^u \frac{b(\tau, v-(u-\tau))\cdot F^c(v)\cdot f(v)}{F^c(v-(u-\tau))\cdot F^c(v)}dvdu \\
&= \int_x^\infty \int_0^\tau \frac{b(\tau,z)\cdot f(z+u-\tau)}{F^c(z)}dzdu \qquad (z = v-u+\tau) \\
&= \int_0^\tau \left(\int_x^\infty f(z+u-\tau)du\right)\frac{b(\tau,z)}{F^c(z)}dz \\
&= \int_0^\tau b(\tau,z)\frac{F^c(z+x-\tau)}{F^c(z)}dz \\
&= \int_0^\tau b(\tau, z+x-\tau)dz.
\end{aligned} \tag{6.3}$$

Combining (6.2) and (6.3) yields

$$b(x) = \int_0^\tau b(\tau, z+x-\tau)dz \quad \text{for} \quad x > \tau. \tag{6.4}$$

Equation (6.4) allows us to conclude the structure of the busy server fluid density, as specified by (3.12) in the paper.

References

Cay Horstmann. Big java early objects. *Interfaces*, 9(10):10, 2002.

W. Whitt. Approximating a point process by a renewal process, I: two basic methods. *Oper. Res.*, 30:125–147, 1982.