Performance Evaluation 162 (2023) 102381

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Load balancing policies without feedback using timed replicas ☆

Rooji Jinan [a], Ajay Badita [b], Tejas Bodas [c], Parimal Parag [b,*]

[a] Department of cyber–physical systems, IISc, Bengaluru, KA 560012, India
[b] Department of electrical communication engineering, IISc, Bengaluru, KA 560012, India
[c] Computer systems group at IIIT Hyderabad, TS 500032, India

ARTICLE INFO

Keywords:
Load balancing
Redundant computing
Distributed discard policy

ABSTRACT

Dispatching policies such as join the shortest queue (JSQ), join the queue with smallest workload (JSW), and their power of two variants are used in load balancing systems where the instantaneous queue length or workload information at all queues or a subset of them can be queried. In situations where the dispatcher has an associated memory, one can minimize this query overhead by maintaining a list of idle servers to which jobs can be dispatched. Recent alternative approaches that do not require querying such information include the cancel-on-start and cancel-on-complete replication policies. The downside of such policies however is that the servers must communicate either the start or the completion time instant of each service to the dispatcher and must allow the coordinated and instantaneous cancellation of all redundant replicas. In practice, the requirements of query messaging, memory, and replica cancellation pose challenges in their implementation and their advantages are not clear. In this work, we consider load-balancing policies that do not need to query load information, do not need memory, and do not need to cancel replicas. Our policies allow the dispatcher to append a timer to each job or its replica. A job or a replica is discarded if its timer expires before it starts receiving service. We analyze several variants of this policy which are novel and simple to implement. We numerically observe that the variants of the proposed policy outperform popular feedback-based policies for low arrival rates, despite no feedback from servers to the dispatcher.

1. Introduction

Load balancing policies play a vital role in latency reduction in distributed systems such as large data centers and cloud computing. A typical load-balancing system comprises of a large number of homogeneous servers and a dispatcher that routes arriving jobs to the queue of these servers. When the instantaneous queue length of different servers is known, an obvious approach would be to use the join-shortest-queue (JSQ) policy [1]. If instead of queue length, the workload i.e., the pending amount of work at each server is known, the optimal policy is the join smallest work queue (JSW). Unfortunately, in most practical systems, the number of servers is large, and therefore obtaining the instantaneous queue lengths or workloads from all servers is difficult.

☆ The research of Parimal Parag was supported in part by the Qualcomm 6G University Relations India, supported by the Qualcomm Inc.; in part by the IBM-IISc Hybrid Cloud Lab (IIHCL) open research collaboration, supported by the IBM India Research Lab; in part by the Robert Bosch Centre for Cyber-Physical Systems; and in part by the Centre for Networked Intelligence (a Cisco Corporate Social Responsibility (CSR) Initiative) at Indian Institute of Science, Bengaluru. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

* Corresponding author.
E-mail addresses: roojijinan@iisc.ac.in (R. Jinan), ajaybadita@iisc.ac.in (A. Badita), tejas.bodas@iiit.ac.in (T. Bodas), parimal@iisc.ac.in (P. Parag).

https://doi.org/10.1016/j.peva.2023.102381
Received 14 October 2022; Received in revised form 5 October 2023; Accepted 6 October 2023
Available online 11 October 2023

A popular remedy for this is to consider the power of d choice variant of JSQ and JSW. In a JSQ(d) policy, the dispatcher samples d servers uniformly at random and queries their queue lengths. The job is then routed to a sampled server with the least number of waiting jobs. Implementing such a policy requires $2d$ messages per job and was shown to have very good performance characteristics [2,3]. The equivalent workload-based policy JSW(d) also has a $2d$ query overhead per job and was analyzed recently [4,5]. For many systems, a $2d$ query exchange is considerable overhead, especially when d is large or when the timescale for message exchange is comparable to the actual service requirement of a job [6].

Recent efforts have therefore been directed towards bringing down this overhead using smart feedback techniques [7,8]. The authors of [7] consider a hyper-scalable dispatching scheme where the dispatcher maintains queue length estimates for the different queues and sends an arriving job to the server with the least estimated queue length. Each server occasionally updates the dispatcher about its true queue length and this enables the dispatcher to synchronize its estimates with reality. The authors of [8] introduce the join-open-queue scheme where servers send busy alerts to the dispatcher at predetermined times. When a server is idle, it does not send the alert and thus the dispatcher can infer idle servers without considerable message exchanges. In such cases, there is some feedback communicated by the servers to the dispatcher, and this can be non-negligible in some settings. Furthermore, the dispatcher operates under noisy queue/workload information, which affects the system performance. It is well known that for correlated processes, there is a tradeoff between the estimated accuracy and the frequency of updates [9,10]. Another policy that works under sparse communication and approximate state information can be found in [11].

The feedback communication overhead and the noisiness of estimates get exacerbated in the case of multiple dispatchers, which is common for modern data centers comprising of a huge number of servers. A load balancing system with multiple dispatchers is analyzed in [12,13], where the authors consider policies that require infrequent communication between servers and dispatchers. In these policies, the dispatchers perform load balancing based on a local estimate of the queue length. It is observed that in such systems, jobs could be concurrently dispatched by different dispatchers to the same server which might drive the system to instability.

An alternative low-feedback policy that uses memory, is the join-idle-queue (JIQ) policy. In this policy, idle queues willingly inform the dispatcher about their idleness and the dispatcher lists this in an associated memory. This policy records accurate information on the idleness of all queues and has very good performance characteristics [14]. An arriving job is sent to an idle queue selected randomly from the list of empty queues and therefore this policy has an overhead of a single feedback message in each busy period per server. Some recent load balancing policies that make use of memory in their dispatching decisions appear in [15,16].

An alternative way to achieve good performance without querying instantaneous queue length or workload information is to use redundancy-based load balancing policies. Two popular variants of redundancy-d based load balancing are cancel-on-start (c.o.s.) [17] and cancel-on-complete (c.o.c.) [18]. In these policies, independent replicas of an arriving job are sent to d randomly chosen servers. In c.o.s. (resp. c.o.c.), when one of the copies starts receiving service (resp. completes service), the $d-1$ replicas are canceled. Such policies also have superior delay performance and are quite amenable to analysis. A detailed product form analysis characterizing the delay performance for both variants is presented in [19,20]. However, a major implementation problem with replication-based policies is the synchronized cancellation of the redundant replicas. The sophistication required for implementing such an approach in fact may even be non-trivial. Further, depending on the operating scenario, instantaneous cancellation may not always be feasible, thereby adding overhead to the system [21,22]. In many redundancy-based practical systems, replica cancellation is an undesirable overhead that is often avoided. The authors of [23,24] discuss applications where delay due to request cancellations cannot be tolerated and describe systems where it is difficult to incorporate functionality to terminate requests while being executed. The authors of [25–27] study systems where replication is implemented without cancellation (r.w.c.). In particular, the authors of [25] suggest that there is a threshold system load above which replication can be detrimental and cannot offer any improvement in terms of latency. This provides the motivation for designing a policy that would intelligently replicate only when the workload conditions are favorable. The idea of replication without cancellation has also been used in multipath routing in networks [28,29].

Besides replication-based policies for latency reduction, latency in distributed storage systems with maximum distance separable (MDS) coded data has been widely studied in literature [30–35]. Although they have superior delay performance, such schemes have additional decoding costs and scalability issues besides the cancellation costs. Also, there are efficient replication-based strategies that have competitive performance with that of MDS coded systems [36,37] and we do not discuss them in this article. In addition, load balancing policies that consider different cost functions like throughput [38], server utilization [39–41] etc. have also been studied. We do not delve into these details.

Note that except for the r.w.c. policy, the load balancing policies discussed earlier either involve (a) communication of messages, or (b) require a memory, or (c) require replication with cancellation. Such policies therefore always have an element of feedback from the server to the dispatcher. In this work, we aim to characterize the impact of such communication/memory/coordinated replica-cancellation in load balancing by comparing their performance with policies that do not need server feedback information. In particular, we focus on static load balancing policies that do not demand queue length information or memory at the dispatcher. The static load balancing approach is very similar to the forward error correction [42, Chapter 1] in communication, where the message redundancy is designed in advance without any receiver feedback. Analogously, the traditional server feedback-based load balancing approaches are similar to adaptive coding policies [42, Chapter 22].

While we allow the dispatcher to possibly replicate jobs to d different servers, we assume that the dispatcher does not have any server state information and does not send any state-dependent cancellation message. This is in line with some of the practical policies discussed in the preceding paragraph. While the random routing policy ($d = 1$) is an obvious choice for such static load balancing, its performance is known to be poor and is therefore of limited interest. Replicate-without-cancellation (r.w.c.) is an alternative candidate, but the impact on the system load due to uncanceled replicas is not clear. At this point, an imminent question is

can we add some functionality to the system (without incurring much overhead) and achieve much better performance as compared to random routing or r.w.c.? Further, would it be possible to have comparable or even better performance as compared to policies like JSQ(d) or JSW(d) that make use of the server state information? An affirmative answer to the latter question even under say a restricted parameter setting, may go a long way in establishing the true value of feedback information in load balancing.

In this paper, we propose a policy where the dispatcher has the ability to append a *server-side cancellation criteria* to each job or its replica. Before picking any job or its replica for service, each server will check if the appended criteria are satisfied or not. If the criteria are met, then the replica is served or else it is dropped. We consider a criterion that depends on the waiting time of the replica in a queue. For example, the criteria that we consider is to serve the replica only if it has waited in the queue for no more than a fixed preset amount of time. More formally, we assume that each arriving job is referred to as the primary replica, and the dispatcher creates $d-1$ secondary replicas. The servers where the replicas are sent are chosen randomly. Associated with the primary and secondary replicas are non-negative *discard thresholds* T_1 and T_2. A replica is discarded by the server if the waiting time experienced by the replica is more than its discard threshold and we label our load balancing policy by $\pi(d, T_1, T_2)$. Such a criterion is easy for the server to validate and can be achieved by logging the arrival time information of each job/replica. Furthermore, our policy can even be implemented in a multiple dispatcher setting without incurring delay overhead and can be designed to not cause instability. The key essence of our approach is to exploit possible gains from the replication of jobs, but at the same time prevent overloading the system due to extra replicas by preemptively performing server-side cancellation of potentially wasteful replicas. We compare this policy against policies with access to side information on the status of the system and show that the proposed load balancing policy in certain regimes provides a superior latency performance as compared to most of the popular policies. This begs two important questions:

1. Can a load balancing policy without server feedback perform as well as the ones with server feedback in certain operating regimes?
2. Are the load balancing policies with server feedback information utilizing the available information optimally?

One needs to answer these two questions to know whether the performance gains derived from using server feedback information are worth the cost structure imposed by such information gathering in the system. We note that answering the first question will need a more thorough study on the characterization of the value of information in such systems and can be an independent study of its own. However, our work provides evidence to show that the utilization of the server feedback information is sub-optimal in many of the prominent load-balancing policies with information feedback. Ideally, if the information utilization is optimal, then the load-balancing policies with extra feedback information are never supposed to perform worse than policies without feedback with the same amount of redundancy and under similar system settings. In this work, we have been successful in designing a policy without server feedback which is numerically shown to outperform the policies with feedback in certain load regimes.

We observe that when T_1 and T_2 are both finite, arriving jobs could potentially be lost without service. Keeping this in mind, the two key performance metrics that we consider are the conditional mean response time of jobs admitted into the system and the loss probability of an arriving job. Note that for systems where loss cannot be tolerated, we can set $T_1 = \infty$ and adapt suitably. To analyze our policy, we make use of the cavity process method of [43,44] along with an assumption on the asymptotic independence of the stationary workloads at the different queues as the number of servers $N \to \infty$. While we prove that the queues are asymptotically independent over any finite time horizon, the absence of monotonicity arguments makes it difficult to extend this result to time-stationary regimes when thresholds T_1, T_2 are finite. When both the discard thresholds are infinite, the workload monotonicity of queues continues to hold, and the asymptotic independence for stationary workloads is easy to prove. Note that asymptotic independence is difficult to prove in general, and proofs are available only under specific service disciplines, load balancing policies, and assumptions on service distributions [43–45]. Having said that, the use of this assumption as a conjecture is widespread [22,43,46] and supported by extensive numerical evidence.

1.1. Key contributions

We have listed our key contributions below.

1. We propose a distributed load balancing policy $\pi(d, T_1, T_2)$, where the dispatcher needs no feedback from the servers. Further, replicas are discarded at a server if the waiting time exceeds the discard threshold.
2. We show that the workloads in the various queues in the system are asymptotically independent over any finite time horizon. We empirically verify that the independence assumption on the limiting marginal workload distribution is a good approximation even for a finite number of servers.
3. To analyze the proposed load balancing policy $\pi(d, T_1, T_2)$, we derive the expressions for key performance metrics such as the loss probability P_L in Lemma 6, the conditional mean response time τ for admitted jobs in Theorem 7, and the moment generating function (MGF) for the limiting workload distribution of an arbitrary queue under the policy $\pi(d, T_1, T_2)$. Furthermore, we invert this function for an exponential service time distribution, to obtain the limiting workload distribution in Corollary 10.
4. We analytically show in Lemma 18 that the proposed policy $\pi(d, \infty, 0)$ always outperforms the random routing policy under exponential service times. We also provide an analytical comparison with the c.o.c.(d) policy when service times are exponential in Proposition 20.

5. We conduct numerical experiments to show that the $\pi(d, \infty, 0)$ policy can outperform the c.o.s.(d), JSQ(d), and JIQ(d), in a low arrival rate regime. This policy converges to the c.o.s.(1) policy in the high arrival rate regime. We observe that the arrival rate threshold for this switch in regime increases with redundancy d.
6. We also provide the performance comparison of our policy with other server feedback-based policies for general service time distributions and observe similar performance improvement as seen under exponential service times.

1.2. Organization

We introduce the system model and notations in Section 2. This is followed by a discussion on the cavity process method and its application to our problem along with a discussion on the asymptotic independence of the workloads at different queues. In Section 3, we compute the performance metrics for the proposed policy $\pi(d, T_1, T_2)$ for a general service time distribution, in terms of limiting marginal workload distribution. In Section 4, we find the closed-form expression for marginal workload distribution when the service time distribution is exponential. We also compute the conditional mean of response time for admitted jobs, for some special cases of $\pi(d, T_1, T_2)$ policy. We provide a comparison of our policy with policies with feedback for various service time distributions in Section 5. We conclude with a summary of our work and future directions in Section 6.

2. System model and preliminaries

We consider a load balancing system with N servers, where jobs arrive according to a Poisson process of rate λN. There is a dispatcher associated with this system whose objective is to minimize the response time experienced by each job by suitably balancing the workload across different servers. Owing to the popularity of redundancy-based load-balancing policies, we assume that the dispatcher has the ability to replicate an arriving job across multiple servers.

Throughout this article, we denote the set of first n consecutive positive integers as $[n] \triangleq \{1, \dots, n\}$, the set of non-negative integers as $\mathbb{Z}_+$, the set of positive integers as $\mathbb{N}$, the set of non-negative reals as $\mathbb{R}_+$ and the set of positive reals as $\mathbb{R}^+$. We also use the notation $x \wedge y \triangleq \min\{x, y\}$.

2.1. Service

We denote the service time for nth arriving job at ith server by $X_{n,i} \in \mathbb{R}_+$. We assume that the random job service time sequence $(X_{n,i} \in \mathbb{R}_+ : n \in \mathbb{N}, i \in [N])$ is independent and identically distributed (*i.i.d.*) with an exponential distribution $G : \mathbb{R}_+ \to [0, 1]$ defined for all $x \in \mathbb{R}_+$ as $G(x) \triangleq (1 - e^{\mu x})\mathbb{1}_{\{x \geqslant 0\}}$, such that mean $\mathbb{E}X_{n,i} = \frac{1}{\mu}$ for each job n and server i. The motivation for independent random service time for each replica at all servers comes from the uncertainties in the time taken to service a job at any server due to other independent background processes [47,48]. The identical distribution models the homogeneity of servers with identical configuration and compute power. Recent studies suggest that the service times in distributed computing systems can be modeled to have two components; a constant startup delay and a random memoryless component [49–52]. Whenever the startup time is negligible the service time distribution can be approximated by an exponential distribution. This along with analytical tractability motivated us to assume that the service times follow *i.i.d.* exponential distribution with rate μ. We denote the tail distribution of the service time or the complementary service time distribution by $\bar{G} \triangleq 1 - G$. When we focus on a single queue i, we will drop the subscript i for brevity.

An alternative and more generalized service model is the S&X model [53] where the service time of nth job at server i is defined to be the product random variable $Y_{n,i} \triangleq S_i \cdot X_n$ where S_i is the slowdown factor at the server i and X_n denotes the random size of the incoming job n. The random slowdown factor S_i is assumed to have a mean greater than or equal to 1. The random job size sequence is assumed to be *i.i.d.* across the jobs. Owing to the difficulty in the analysis posed by this model (also noted in [22,53]), we focus on *i.i.d.* service time model in this work. The *i.i.d.* exponential service time model can be considered as a special case of the S&X model where the slow down factor S is *i.i.d.* exponential with unit rate across servers and the service times of job X_i have a constant size of $\frac{1}{\mu}$. Another interesting special case of S&X model is when the slowdown factor is deterministic and the job sizes are *i.i.d.* . This special case has been discussed in detail in Appendix E.

2.2. Threshold based cancellation

We assume that the dispatcher has limited functionality and that it cannot cancel redundant copies when one of the replicas has received (or started receiving) service. Instead, we assume that the dispatcher can append *discard instruction* along with each replica. Before a job/replica starts service, each server will read the *discard instruction* and possibly discard the replica based on the instruction. We call this a redundancy-based approach with server-side cancellation of replicas. For ease of exposition, we assume that the *instruction* is almost identical for all copies in the system and hence the overhead of implementing this approach is minimal. In this article, we restrict to *instructions* that are characterized by a threshold $T \in [0, \infty)$. In particular, we assume that the server serves a replica if it is chosen for service within T units of its arrival or else discards the replica. We call T as the *discard threshold* for brevity.

2.2.1. Primary replica and discard threshold

We consider the following dispatching policy based on the above idea of a *discard threshold*. When a job arrives, the dispatcher samples a single *primary* server uniformly at random and sends a primary replica of the job to the server along with the *primary discard threshold* T_1.

2.2.2. Secondary replicas and discard thresholds

For each job arrival, the dispatcher creates $d-1$ secondary replicas, samples $d-1$ other servers uniformly at random, and sends each of the secondary replicas to the sampled $d-1$ servers after appending each replica with a *secondary discard threshold* of T_2 where $T_2 \leqslant T_1$. We choose the secondary discard threshold to be smaller than the primary discard threshold to ensure that the secondary replicas will not overload the system when the current workloads at the queues are high. Furthermore, note that we expect the secondary replicas to be helpful only if the primary is delayed.

Since our policy is parametrized by the number of replicas d, primary discard threshold T_1, and secondary discard threshold T_2, we shall henceforth denote the proposed policy by $\pi(d, T_1, T_2)$ for simplicity. Following are some special cases of our *discard threshold* based redundancy-d policy that we analyze in this article.

1. Replication with identical thresholds, $\pi(d, T, T)$: In this policy, each job is replicated d times and assigned to d servers chosen at random. Each job replica will have a threshold of T time units which can possibly result in a loss of jobs. When $T = \infty$, the policy reduces to that of a simple replication-d policy without cancellation.
2. Replication with no loss, $\pi(d, \infty, T_2)$: Under this policy, as the primary threshold $T_1 = \infty$, each primary replica of the job is definitely served. The advantage of this policy is that no jobs are lost.
3. Replication on idle secondary servers, $\pi(d, \infty, 0)$: This is a special case of replication policy with minimal redundancy addition since secondary replicas only join idle queues.

2.3. Server

We assume that each server has an infinite-sized buffer where arriving job replicas can wait for service, on a first come first served (FCFS) basis. We let the random variable $W_{n,i}$ denote the waiting time for the nth arriving job at server $i \in [N]$. Due to FCFS service, the random variable $W_{n,i}$ is also the effective workload present at server i that must be served before nth job replica can receive service. An arriving replica is executed at a server i if its discard threshold T is larger than the observed workload $W_{n,i}$, and is discarded otherwise.

Each arriving job in the system results in a potential arrival at a maximum of d randomly sampled queues. Depending on the discard threshold T and waiting time $W_{n,i}$, the job either receives service or is discarded. If a replica is served, then it results in an actual arrival at the corresponding server queue.

Definition 1. For the nth arriving job, let $I_{n,1}$ be the singleton set of servers where the primary replica is dispatched, and $I_{n,2}$ be the set of servers to which the secondary replicas are dispatched. For the job n, whether server j is selected for service of a primary or secondary replica is denoted by indicators $\gamma^1_{n,j} \triangleq \mathbb{1}_{\{j \in I_{n,1}\}}$ and $\gamma^2_{n,j} \triangleq \mathbb{1}_{\{j \in I_{n,2}\}}$ respectively.

Definition 2. If a replica for job n is dispatched to a server $j \in I_{n,1} \cup I_{n,2}$ with current workload $W_{n,j}$, then we define the indicator that the job is not discarded at this server j as

$$\xi_{n,j} \mathbb{1}_{\{j \in I_{n,1} \cup I_{n,2}\}} \triangleq \mathbb{1}_{\{W_{n,j} \leqslant T_1\}} \gamma^1_{n,j} + \mathbb{1}_{\{W_{n,j} \leqslant T_2\}} \gamma^2_{n,j}. \tag{1}$$

We denote the set of servers where the replicas for job n are not discarded by $I_n \triangleq \left\{ j \in I_{n,1} \cup I_{n,2} : \xi_{n,j} = 1 \right\}$. A job is not discarded when $I_n \neq \emptyset$ and we denote this by indicator $\xi_n \triangleq \mathbb{1}_{\{I_n \neq \emptyset\}}$. We can write this in terms of the set of servers $I_{n,1}, I_{n,2}$, the indicator $\xi_{n,j}$, and its complement $\bar{\xi}_{n,j} \triangleq 1 - \xi_{n,j}$ for all $j \in I_{n,1} \cup I_{n,2}$,

$$\xi_n \triangleq 1 - \prod_{j \in I_{n,1}} \bar{\xi}_{n,j} \prod_{j \in I_{n,2}} \bar{\xi}_{n,j}. \tag{2}$$

From the symmetry in the system, the marginal workload for each server j has an identical distribution. We denote the limiting marginal workload distribution for any server $j \in [N]$ by $F : \mathbb{R}_+ \to [0, 1]$, such that $F(x) \triangleq \lim_{n \to \infty} P\left\{ W_{n,j} \leqslant x \right\}$ for $x \in \mathbb{R}_+$, when the limit exists. Since F is the limiting marginal workload distribution seen by an arriving customer, it follows from the PASTA property that F is also the stationary distribution of marginal workload in the system.

2.4. Performance metrics

We consider the following two stationary performance metrics, the limiting mean response time and the limiting loss probability. Since our dispatcher replicates each arriving job to at most d servers, the response time of an arriving job is the minimum of the sojourn times experienced by its different replicas. When both the thresholds T_1 and T_2 are finite, each replica can be discarded without service, leading to a loss. For lost jobs, the response time metric is meaningless. Hence, we obtain the mean response time of a job, conditioned on the event that it is not discarded. A job is serviced when at least one of its replicas is not discarded at the servers sampled by the dispatcher, i.e. when the workload at one of these servers is smaller than or equal to the corresponding discard threshold.

Definition 3. The limiting loss probability for policy $\pi(d, T_1, T_2)$ is denoted by $P_L \triangleq \lim_{n\to\infty} \mathbb{E}\xi_n$.

Definition 4. We denote the response time of nth job by $R'_n \in \mathbb{R}_+ \cup \{\infty\}$ and the response time of this job if it is not discarded job by $R_n = \xi_n R'_n \in \mathbb{R}_+$. We denote the distribution function for an undiscarded job at stationarity by $H : \mathbb{R}_+ \to [0,1]$ such that for all $x \in \mathbb{R}_+$

$$H(x) \triangleq \lim_{n\to\infty} P\left\{R_n \leqslant x\right\}.$$

The tail distribution $\bar{H} : \mathbb{R}_+ \to [0,1]$ is defined as $\bar{H}(x) \triangleq 1 - H(x)$ for all $x \in \mathbb{R}_+$. We study the conditional mean response time for a job given that it is not discarded, which is defined as

$$\tau \triangleq \frac{\lim_{n\to\infty} \mathbb{E}[R_n]}{\lim_{n\to\infty} \mathbb{E}[\xi_n]} = \frac{\int_{x\in\mathbb{R}_+} \bar{H}(x)dx}{1 - P_L}. \tag{3}$$

In this article, we analyze the performance of the $\pi(d, T_1, T_2)$ load balancing policy for different special cases mentioned in Section 2.2, based on the two performance metrics of conditional mean response time and loss probability. Computing the limiting marginal workload distribution at a single queue is straightforward and can be performed by isolating the considered queue from the rest of the system. However, a job response time is the minimum response time for all possible job replicas, and computation of the conditional mean requires the knowledge of the joint distribution of workloads at all queues with a job replica. We point out that the workloads at different queues are not independent of each other due to the correlated arrivals. We illustrate the workload dependence at different servers in the following example.

Example 1. Consider a system of two servers with initial workloads $W_1(0) = W_2(0) = 0$. Suppose the job arrival process to the system is Poisson with the homogeneous rate λ, and each arriving job has a constant size c. The jobs are replicated and sent to both servers and they are accepted at the servers if their current workloads are smaller than a threshold of T. We denote the inter-arrival times by the random sequence $(Z_n \in \mathbb{R}_+ : n \in \mathbb{N})$, and the nth arrival instant by $S_n \triangleq \sum_{k=1}^{n} Z_k$ for all $k \in \mathbb{N}$. Then the workload at server i at the nth arrival instant is denoted by $W_{n,i}$, and can be written recursively, as $W_{n+1,i} = (W_{n,i} + c - Z_{n+1})_+$. We observe that $W_{n,1} = W_{n,2}$ for all $n \in \mathbb{N}$. Further, we have $W_i(t) = (W_{n,i} - t)_+$ for all $t \in [S_n, S_{n+1})$, and hence $W_1(t) = W_2(t)$ for all $t \in \mathbb{R}_+$. We observe that the workloads in these two queues are completely identical at all times, and not independent of each other.

However, we show that for the proposed load balancing policy $\pi(d, T_1, T_2)$, the workloads at different queues are independent of each other for any finite time horizon $[0, t]$ when the job arrival process is homogeneous Poisson, the replicas have *i.i.d.* service time distribution, and the number of servers N grows large while keeping the number of replicas d fixed. Furthermore, to compute the joint workload distribution, we use the cavity process method [4,22,43,44] that assumes the asymptotic independence of the limiting marginal workload at server queues. The next subsection provides a brief discussion of the cavity process method.

2.5. Cavity process method

Here, we explain the principle of a cavity process method as applied to popular load balancing policies such as least loaded (JSW(d)) or join-shortest-queue (JSQ(d)) and then specialize the discussion to our policy $\pi(d, T_1, T_2)$. See [4,22,43,44] for more details about this approach. In the JSW(d) (resp. JSQ(d)) system with N queues and Poisson arrival rate of λN, d queues are sampled for each arriving job. The arriving job is executed on the sampled server with the smallest workload (resp. queue length). Let $\{\mathcal{H}(t), t \geqslant 0\}$ denote the collection of probability measures on $\mathbb{R}_+$. This is called the environment process. We tag one of the queues in the N queue system as the cavity queue and denote the cavity process by $(X^{\mathcal{H}(t)}, t \geqslant 0)$ which represents the workload process (resp. the queue length process) at the cavity queue under policy JSW(d) (resp. JSQ(d)). The potential arrival rate of jobs to the cavity queue under both policies is λd. For a potential arrival at the cavity queue at time t, we compare $d-1$ random variables with law $\mathcal{H}(t)$ and cavity random variable $X^{\mathcal{H}(t-)}$. The potential arrival becomes an actual arrival to the cavity queue if the value of $X^{\mathcal{H}(t-)}$ is lower than the values taken by the $d-1$ other variables, else the job is discarded. When the job is accepted, we have $X^{\mathcal{H}(t)} = X^{\mathcal{H}(t-)} + 1$ for the JSQ(d) policy and $X^{\mathcal{H}(t)} = X^{\mathcal{H}(t-)} + x$ for the JSW(d) policy where x is the service requirement of the arriving job. When the job is discarded, we have $X^{\mathcal{H}(t)} = X^{\mathcal{H}(t-)}$. For the JSW($d$) policy, the workload $X^{\mathcal{H}(t)}$ at the cavity queue decreases at a unit rate, and for the JSQ(d) policy, the queue length $X^{\mathcal{H}(t)}$ of the cavity queue decreases by one at a unit rate. The process $\mathcal{H}(\cdot)$ is called as the *equilibrium environment process* if $X^{\mathcal{H}(\cdot)}(t)$ has distribution $\mathcal{H}(t)$ for all times t. If $\mathcal{H}(t) = \mathcal{H}$ for all t, then $\mathcal{H}$ is called as *equilibrium environment*.

The cavity process method was used in [43,44] to analyze the JSW(d) and the JSQ(d) policy. A key step in the analysis is to show asymptotic independence between the workloads/queue length random variables at the different queues. While the analysis for JSW(d) holds for any service requirement distribution, the proof for JSQ(d) is only known for the case when the service requirement of a job has a decreasing hazard rate distribution. In [4], this approach is used further to obtain the functional differential equation for the workload distribution of the cavity queue. In [22], several workload-based load-balancing policies based on redundancy were considered and the cavity process method was used to identify the workload distribution for a wide range of load-balancing policies. While the asymptotic independence of the queues was only conjectured, this was very recently proved in [45] for a variety of such replication-based policies, including most of the policies of [22]. Asymptotic independence of workloads at stationarity requires a change of limits, which is shown to hold true under the system monotonicity properties in [45]. We prove the asymptotic

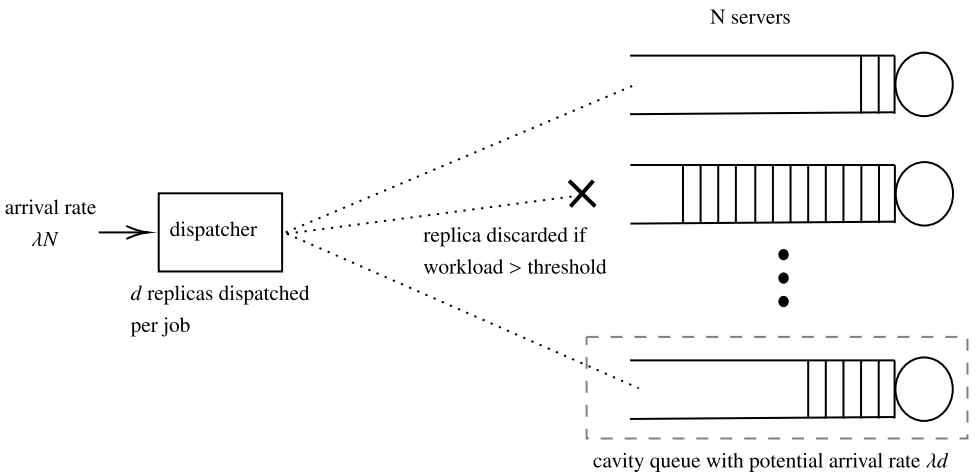


Fig. 1. The N server system under $\pi(d, T_1, T_2)$ policy with a job arrival rate λN. The dispatcher dispatches d replicas per job and the potential arrival rate at any cavity queue is λd.

independence of workloads under our settings for any finite time horizon in Proposition 5. However, since the proposed load-balancing policy does not satisfy monotonicity properties, we are only able to provide empirical validation at time stationarity.

For the proposed $\pi(d, T_1, T_2)$ policy, we use this cavity process method along with the conjecture that the workload distribution across any finite subset of queues is asymptotically independent. For the proposed policy shown in Fig. 1, we note that the potential arrival rate to the cavity queue is $\tilde{\lambda} \triangleq \lambda d$. If the copy at the cavity queue is a primary replica, then $X^{\mathcal{H}(t)} = X^{\mathcal{H}(t-)} + x$ if $X^{\mathcal{H}(t-)} \leqslant T_1$ else the copy is discarded. Similarly, if the replica at the cavity queue is a secondary one, then the replica is served if $X^{\mathcal{H}(t-)} \leqslant T_2$. Clearly, the potential arrival at the cavity queue becomes an actual arrival based on the workload level at the queue. Remarkably, for our policy, there is no influence on the cavity queue of the $d-1$ random variables with law $\mathcal{H}(\cdot)$. With the assumption of the asymptotic independence of the workload at finitely many queues, and using the cavity process approach, we can view the cavity queue as an $M/G/1$ queue with workload-dependent arrival rates. The workload distribution of the cavity queue is in fact the equilibrium environment $\mathcal{H}$ for our system. See [54] for one possible approach to obtain the workload distribution for an $M/G/1$ queue with workload-dependent arrival rates. In the following, we use a different approach based on the Lindley type recursion and moment generating function (MGF) to obtain the workload distribution for the queue at the cavity. We believe that this approach is novel and can be applied to more general load-balancing policies beyond this work.

Next, we discuss the conjecture on asymptotic independence. First, we provide the result on asymptotic independence of workloads over a finite horizon. The proof is very similar to the proof of asymptotic independence of queues over a finite time horizon for JSQ(d) dispatch policy, shown in [44, Proposition 7.1].

Proposition 5 (*Asymptotic Independence Over Finite Time Horizon*)**.** *Consider an N server system under $\pi(d, T_1, T_2)$ dispatch policy. When the number of servers N grows asymptotically large, the marginal workload distributions at any finite number of queues are independent over a finite time horizon.*

See Appendix A for proof.

Remark 1. The above proposition holds true for general *i.i.d.* service time distributions as well as for the identical service time model discussed in Appendix E. For the ease of exposition, we omit the details.

For the power-of-d variants of dispatch policies, once the asymptotic independence is shown for a finite time horizon, one can show the asymptotic independence at time stationarity as well when the workloads satisfy certain monotonicity conditions. (See Appendix A for further details.) Although we do not have such monotonicity property under $\pi(d, T_1, T_2)$ policy when either of the two thresholds T_1, T_2 are finite, we conjecture that the asymptotic independence of server workloads continues to hold true at time stationarity.

Conjecture 1 (*Asymptotic Independence at Stationarity*)**.** *Consider an N server system under $\pi(d, T_1, T_2)$ dispatch policy. When the number of servers N grows asymptotically large, the system has a unique equilibrium workload distribution under which workloads at any finite number of queues are independent.*

See Appendix B for empirical validation of this conjecture. Please note that Conjecture 1 can be proved for a limited regime of arrival rates, by adapting the proof of [44, Theorem 2.3] to our setting. However, the empirical evaluations suggest that asymptotic independence is a valid assumption under all arrival rates under the studied policy.

Remark 2. We first obtain the MGF for the workload at the cavity queue. We then use this to obtain the conditional mean response time for the different policies, under Conjecture 1. We illustrate the accuracy of our expressions in Appendix B by comparing them with simulation experiments for different values of N. As a validation of the assumption, we see that as N increases, the mean response time from simulations approaches the analytical values.

3. Performance analysis

As mentioned before, we can compute the identical marginal workload distribution at all N servers for the proposed $\pi(d, T_1, T_2)$ dispatch policy. However, the expressions for both the performance metrics of conditional mean response time and loss probability, can only be obtained under Conjecture 1. This computation is an approximation for a finite number of servers. However, we empirically verify that this approximation is quite accurate even for a small number of servers.

3.1. Loss probability

When both primary and secondary thresholds are finite, some jobs can be discarded from the system. Under Conjecture 1, we compute the limiting loss probability of a job being discarded in the following Lemma.

Lemma 6. *The limiting loss probability of a job under $\pi(d, T_1, T_2)$ dispatch policy with equilibrium workload distribution F and tail distribution of service time $\bar{G}$ is given by $P_L = \bar{F}(T_1)\bar{F}(T_2)^{d-1}$.*

Proof. From (2) in Definition 3, we obtain $P_L = \lim_{n\to\infty} \mathbb{E}\left[\prod_{j\in I_{n,1}} \bar{\xi}_{n,j} \prod_{j\in I_{n,2}} \bar{\xi}_{n,j}\right]$. The result follows from the independence of the indicators $(\bar{\xi}_{n,j} : j \in I_{n,1} \cup I_{n,2})$, under the assumption of asymptotic independence of workloads $W_{n,j}$ across the servers $j \in I_{n,1} \cup I_{n,2}$ when $n \to \infty$, and the fact that the mean of indicators are $\mathbb{E}[\bar{\xi}_{n,j} \mathbb{1}_{\{j\in I_{n,1}\cup I_{n,2}\}} \Big| I_{n,1}, I_{n,2}] = \bar{F}(T_1)\gamma^1_{n,j} + \bar{F}(T_2)\gamma^2_{n,j}$. □

3.2. Conditional mean response time

Note that when the discard thresholds T_1, T_2 are finite, then all jobs that arrive at a server with workload $w > T_1$ will be lost. For lost jobs, the response time metric is meaningless. Hence, we obtain the conditional mean response time given that the job is not discarded. A job is serviced when at least one of its replicas is not discarded at the servers sampled by the dispatcher, i.e. when the workload at one of these servers is smaller than or equal to the corresponding discard threshold.

Theorem 7. *The conditional mean response time of an undiscarded job under $\pi(d, T_1, T_2)$ policy with equilibrium workload W having distribution F and tail distribution of service time $\bar{G}$, is given by*

$$\tau = \frac{1}{1-P_L} \int_x \left[(\bar{F}(T_1) + k(x, T_1))(\bar{F}(T_2) + k(x, T_2))^{d-1} - \bar{F}(T_1)\bar{F}(T_2)^{d-1} \right] dx,$$

where $k(x, T) \triangleq \mathbb{E}\left[\bar{G}(x - W)\mathbb{1}_{\{W\leqslant T\}}\right]$ is the tail distribution of sojourn time for an undiscarded job at stationarity with discard threshold T.

Proof. Refer to Appendix C. □

Remark 3. To understand the mean conditional response time, we need to understand the tail distribution $k(x, T)$. Exchanging integral and expectation from the monotone convergence theorem for non-negative functions, we observe that the mean sojourn time for an undiscarded job is

$$\int_{x\in\mathbb{R}_+} k(x, T)dx = \mathbb{E}\left[\mathbb{1}_{\{W\leqslant T\}}\left(\int_0^W dx\bar{G}(x - W) + \int_W^\infty dx\bar{G}(x - W)\right)\right] = \mathbb{E}\left[W\mathbb{1}_{\{W\leqslant T\}}\right] + \frac{F(T)}{\mu}.$$

Defining $k(x) \triangleq \lim_{T\to\infty} k(x, T)$ for all $x \in \mathbb{R}_+$, we observe that $\int_{x\in\mathbb{R}_+} k(x)dx = \mathbb{E}W + \frac{1}{\mu}$ is the mean sojourn time of any arriving customer at stationarity. Since $W\mathbb{1}_{\{W\leqslant T\}} \leqslant T\mathbb{1}_{\{W\leqslant T\}}$, we get

$$\int_{x\in\mathbb{R}_+} k(x, T)dx \leqslant \mathbb{E}W \wedge (TF(T)) + \frac{F(T)}{\mu}. \tag{4}$$

Remark 4. For a single primary replica $d = 1$, the loss probability $P_L = \bar{F}(T)$ and the conditional mean response time for admitted jobs is $\tau = \frac{1}{\mu} + \frac{\mathbb{E}\left[W\mathbb{1}_{\{W\leqslant T\}}\right]}{F(T)} \leqslant \frac{1}{\mu} + T$.

Remark 5. When the thresholds T_1 and T_2 are infinity, we see the tail workload distributions $\bar{F}(T_1) = \bar{F}(T_2) = 0$ and we have $k(x) = k(x, \infty) = \mathbb{E}\bar{G}(x - W)$. It follows that the tail distribution of response time is $\bar{H}(x) = k(x)^d$.

Remark 6. Since workload $W \geqslant 0$, we have $k(x, 0) = \bar{G}(x)$. Thus, for the thresholds $T_1 = \infty$ and $T_2 = 0$, the loss probability is zero and the tail distribution of response time is $\bar{H}(x) = k(x)\bar{G}(x)^{d-1}dx$.

4. Workload distribution and conditional mean response time under exponential service times

In this section, we evaluate the limiting workload distribution F in the cavity queue under various load balancing policies discussed in Section 2.2 when the service time of each job is independent and follows an identical exponential distribution with rate μ. We choose the service times to be exponentially distributed as they are amenable to analytical computations, due to their memoryless property. Let us first introduce some preliminary definitions prior to introducing the results.

Recall that the indicator that the jth server is selected by nth job as a primary or secondary server is $\gamma^1_{n,j}$ or $\gamma^2_{n,j}$ respectively. Further, the workload seen by the nth job arrival at server j is $W_{n,j}$ and the service time for nth job, if it joins server j, is given by $X_{n,j}$. Since we are interested in a single cavity queue j, we drop the subscript j in the following. For $T_2 \leqslant T_1$, we can use Lindley's recursion to write the single queue workload sequence $(W_n : n \in \mathbb{N})$ in terms of random service time sequence $(X_n : n \in \mathbb{N})$ and inter-arrival time sequence $(Z_n : n \in \mathbb{N})$, as

$$W_{n+1} = \left(W_n + X_n \Big((\gamma^1_n + \gamma^2_n) \mathbb{1}_{[0,T_2]}(W_n) + \gamma^1_n \mathbb{1}_{(T_2,T_1]}(W_n) \Big) - Z_{n+1} \right)_+, \quad n \in \mathbb{Z}_+. \tag{5}$$

We define $J_n \triangleq (\gamma^1_n + \gamma^2_n)\mathbb{1}_{\{W_n \in [0,T_2]\}} + \gamma^1_n \mathbb{1}_{\{W_n \in (T_2,T_1]\}}$ as the indicator for a replica to arrive at the server for each arrival $n \in \mathbb{N}$, to re-write Lindley's recursion (5) for the evolution of marginal workload as $W_{n+1} = (W_n + X_n J_n - Z_{n+1})_+$ where $\sigma(J_n) \subseteq \sigma(W_n, \gamma^1_n, \gamma^2_n)$ and the conditional mean $\mathbb{E}[J_n \mid W_n] = \frac{d}{N}\mathbb{1}_{[0,T_2]}(W_n) + \frac{1}{N}\mathbb{1}_{(T_2,T_1]}(W_n)$ for all $n \in \mathbb{N}$. It follows that $(W_n : n \in \mathbb{N})$ is a reflected random walk with step-size sequence $(X_n J_n - Z_{n+1} : n \in \mathbb{N})$ and hence the limiting workload distribution $F(w) = \lim_{n\to\infty} P\left\{W_n \leqslant w\right\}$ exists for all $w \in \mathbb{R}$ if $\mathbb{E}[X_n J_n \mid \left\{W_n = w\right\}] < \mathbb{E}Z_{n+1}$ for all w except in a finite bounded set. In order to derive the stationary workload distribution in the cavity queue, we make use of the moment-generating function of the stationary workload.

Definition 8. The moment-generating function of the limiting workload W in a single queue, restricted to different workload regimes is defined as

$$\boldsymbol{\Phi}_W(\theta) \triangleq \mathbb{E}\left[e^{-\theta W}\right], \quad \boldsymbol{\Phi}_2(\theta) \triangleq \mathbb{E}\left[e^{-\theta W}\mathbb{1}_{\{W>T_2\}}\right], \quad \boldsymbol{\Phi}_1(\theta) \triangleq \mathbb{E}\left[e^{-\theta W}\mathbb{1}_{\{W>T_1\}}\right].$$

Remark 7. We observe that $\boldsymbol{\Phi}_W(\theta) - \boldsymbol{\Phi}_2(\theta) = \mathbb{E}\left[e^{-\theta W}\mathbb{1}_{[0,T_2]}(W)\right]$ and $\boldsymbol{\Phi}_2(\theta) - \boldsymbol{\Phi}_1(\theta) = \mathbb{E}\left[e^{-\theta W}\mathbb{1}_{(T_2,T_1]}(W)\right]$ exists for all $\theta \in \mathbb{R}$, since $e^{-\theta W}$ is bounded in bounded intervals $[0, T_2]$ and $(T_2, T_1]$ for all $\theta \in \mathbb{R}$. Further, this implies that the moment-generating functions $\boldsymbol{\Phi}_W(\theta), \boldsymbol{\Phi}_1(\theta), \boldsymbol{\Phi}_2(\theta)$ converge for the same set of values of θ. We further observe that $\theta \geqslant 0$ is sufficient for the existence of all three moment-generating functions.

Theorem 9. *For an N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, the moment generating function $\boldsymbol{\Phi}_W(\theta)$ for the waiting time of admitted jobs at any queue under $\pi(d, T_1, T_2)$ policy is*

$$F(0)\left(1 + \frac{\bar{\lambda}}{\theta + \mu - \bar{\lambda}}\right) + \left((\mu - \lambda)\bar{F}(T_2) + \lambda\bar{F}(T_1)\right)\left[\frac{e^{-\theta T_2}}{\theta + \mu - \lambda} - \frac{e^{-\theta T_2}}{\theta + \mu - \bar{\lambda}}\right] - \mu\bar{F}(T_1)\left[\frac{e^{-\theta T_1}}{\theta + \mu - \lambda} - \frac{e^{-\theta T_1}}{\theta + \mu}\right], \tag{6}$$

where $F(0) = 1 - \frac{\bar{\lambda}}{\mu} + \left[\frac{\bar{\lambda} - \lambda}{\mu}\bar{F}(T_2) + \frac{\lambda}{\mu}\bar{F}(T_1)\right]$ and $\bar{\lambda} = d\lambda$.

Proof. The detailed proof is in Appendix D. □

Corollary 10. *For an N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, the single queue workload distribution under $\pi(d, T_1, T_2)$ policy is given by*

$$\begin{aligned} F(w) = F(0)\Big(1 + \frac{\bar{\lambda}(1 - e^{-(\mu-\bar{\lambda})w})}{\mu - \bar{\lambda}}\Big) - \mu\bar{F}(T_1)\Big(\frac{(1 - e^{-(\mu-\lambda)(w-T_1)_+})}{\mu - \lambda} - \frac{(1 - e^{-\mu(w-T_1)_+})}{\mu}\Big) \\ + ((\mu - \lambda)\bar{F}(T_2) + \lambda\bar{F}(T_1))\Big(\frac{(1 - e^{-(\mu-\lambda)(w-T_2)_+})}{\mu - \lambda} - \frac{(1 - e^{-(\mu-\bar{\lambda})(w-T_2)_+})}{\mu - \bar{\lambda}}\Big). \end{aligned} \tag{7}$$

Remark 8. From the expression (7) for limiting marginal workload distribution at the cavity queue, we observe that there are three distinct regimes for the distribution. When the workload in a cavity queue lies in the duration $[0, T_2)$, the arrival rate to the queue is $\bar{\lambda}$. The cavity queue behaves like an $M/M/1$ queue with arrival rate $\bar{\lambda}$ and service rate μ, and the marginal workload distribution reduces to

$$F(w) = F(0)\Big(\frac{\mu - \bar{\lambda}e^{-(\mu-\bar{\lambda})w}}{\mu - \bar{\lambda}}\Big), \quad w \in [0, T_2).$$

Similarly, when the workload in a cavity queue lies in the duration $[T_2, T_1)$, the arrival rate to the queue is λ. Accordingly, the behavior of the cavity queue in this region is similar to an $M/M/1$ queue with arrival rate λ and service rate μ. As expected, the marginal workload distribution reduces to

$$F(w) = F(T_2) + \frac{(\mu F(0) - (\mu - \bar{\lambda})F(T_2))}{\mu - \lambda}(1 - e^{-(\mu-\lambda)(w-T_2)}), \quad w \in [T_2, T_1).$$

Since there are no more arrivals to a cavity queue when the workload is larger than the threshold T_1, the workload distribution is expected to decay exponentially with the service rate μ. Unsurprisingly, the marginal workload distribution is

$$F(w) = F(T_1) + \bar{F}(T_1)(1 - e^{-\mu(w-T_1)}), \quad w \geqslant T_1.$$

Next, we study some special cases of the $\pi(d, T_1, T_2)$ policy listed in Section 2.2.

4.1. Replication with identical thresholds

First, we study the system under the replication with identical discard thresholds policy, $\pi(d, T, T)$. Note that as the system allows loss, it is always stable. The next result follows from Corollary 10 by substituting $T_1 = T_2$.

Corollary 11. *For an N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, the limiting marginal workload distribution at the cavity queue at stationarity under $\pi(d, T, T)$ policy, is given by*

$$F(w) = \begin{cases} F(0)\left(\frac{\mu}{\mu-\bar{\lambda}} - \frac{\bar{\lambda}}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})w}\right), & 0 < w \leqslant T \\ F(T) + \frac{\bar{\lambda}}{\mu} e^{\bar{\lambda}T} F(0)(e^{-\mu T} - e^{-\mu w}), & w > T \end{cases}$$

where $F(0) = \left[\frac{(1-\frac{\bar{\lambda}}{\mu})}{1-(\frac{\bar{\lambda}}{\mu})^2 e^{-(\mu-\bar{\lambda})T}}\right] \mathbb{1}_{\{\mu \neq \bar{\lambda}\}} + \frac{1}{\bar{\lambda}T+2} \mathbb{1}_{\{\mu=\bar{\lambda}\}}$ *and* $F(T) = \frac{\mu}{\bar{\lambda}}(1 - F(0))$.

Using the above corollary, we now compute the loss probability and conditional mean response time using Theorem 7.

Corollary 12. *The equilibrium loss probability of a job under discard threshold-based dispatching policy $\pi(d, T, T)$ with equilibrium workload distribution F and tail distribution of service time $\bar{G}$, is given by $P_L = \left(1 - \frac{\mu}{\bar{\lambda}}(1 - F(0))\right)^d$, where probability of zero workload $F(0)$ is given in Corollary 11.*

Remark 9. Note that the effective arrival rate at each cavity queue under replication with identical thresholds policy is $\bar{\lambda}\mathbb{1}_{\{w \leqslant T\}}$. However, as the jobs are discarded as soon as the current workload exceeds the threshold T, the queues remain stable even when the arrival rate to the system exceeds the service rate. In particular, the above results says that for $\bar{\lambda} = \mu$, we get the workload distribution $F(w) = \frac{\bar{\lambda}}{\bar{\lambda}T+2} w \mathbb{1}_{\{0<w\leqslant T\}} + \frac{1-e^{-\bar{\lambda}(w-T)}}{\bar{\lambda}T+2} \mathbb{1}_{\{w>T\}}$ and the loss probability $P_L = \left(\frac{1}{\bar{\lambda}T+2}\right)^d$.

From Theorem 7, we know the conditional mean of limiting response time under $\pi(d, T, T)$ policy is $\tau = \frac{1}{1-P_L} \int_x \left((\bar{F}(T) + k(x, T))^d - \bar{F}(T)^d\right) dx$. Thus, we need the tail distribution $k(x, T)$ of sojourn time for an undiscarded job to evaluate the mean response time of the N server system under the policy $\pi(d, T, T)$.

Lemma 13. *For an N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, we can find the following constants under $\pi(d, T, T)$ policy,*

$$\bar{F}(T) = 1 - F(0)\left[\frac{\mu}{\mu-\bar{\lambda}} - \frac{\bar{\lambda}}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})T}\right], \quad F(0) = \frac{(1-\frac{\bar{\lambda}}{\mu})}{1-(\frac{\bar{\lambda}}{\mu})^2 e^{-(\mu-\bar{\lambda})T}}.$$

The tail distribution $k(x, T)$ of sojourn time for an undiscarded job at stationarity is given by

$$k(x, T) = \begin{cases} F(0) e^{-\mu x} e^{\bar{\lambda}T}, & x \geqslant T \\ F(0)\left(\frac{\mu}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})x} - \frac{\bar{\lambda}}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})T}\right), & x < T. \end{cases}$$

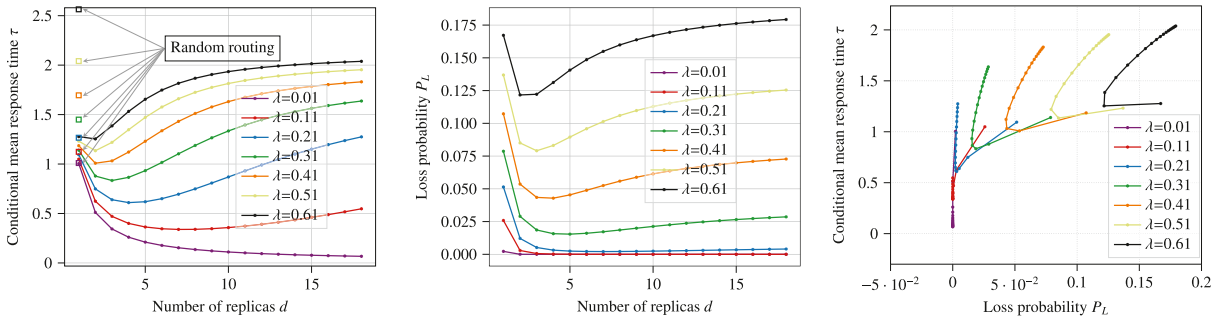
Proof. We know that the service time is exponential and hence the tail service time distribution is $\bar{G}(x) = e^{-\mu(x)_+}$, where $(x)_+ = \max\{x, 0\}$. Therefore, we can write $k(x, T) = \mathbb{E}\left[\bar{G}(x - W)\mathbb{1}_{\{W \leqslant T\}}\right] = F(T) - F(T \wedge x) + e^{-\mu x} \int_0^{T\wedge x} e^{\mu W} dF(w)$. Considering the two cases when $x \geqslant T$ and $x < T$, we get

$$k(x, T) = \begin{cases} e^{-\mu x} \int_0^{T\wedge x} e^{\mu W} dF(w), & x \geqslant T, \\ F(T) - F(x) + e^{-\mu x} \int_0^x e^{\mu W} dF(w), & x < T. \end{cases}$$

The result follows from the workload distribution F given in Corollary 11. □

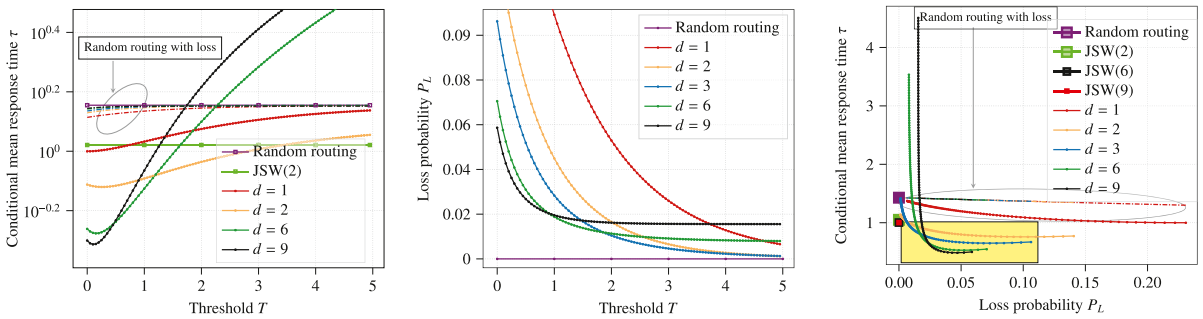
Corollary 14. *For an N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, the limiting loss probability and the conditional mean of limiting response time under $\pi(d, 0, 0)$ policy are given by*

$$P_L = \left(\frac{\bar{\lambda}}{\mu + \bar{\lambda}}\right)^d, \quad \tau = \frac{1}{\mu(\mu + \bar{\lambda})^d} \sum_{i=0}^{d-1} \binom{d}{i} \frac{\bar{\lambda}^i \mu^{d-i}}{(d-i)}.$$



(a) τ vs d. (b) P_L vs d. (c) τ vs P_L.

Fig. 2. We plot the conditional mean response time τ and the loss probability P_L for the policy $\pi(d,T,T)$ as a function of the number of replicas d, for a fixed discard threshold $T = 1.5$, the number of servers $N = 20$, service rate $\mu = 1$, and different values of arrival rate $\lambda \in \{0.01, 0.11, \ldots, 0.61\}$.



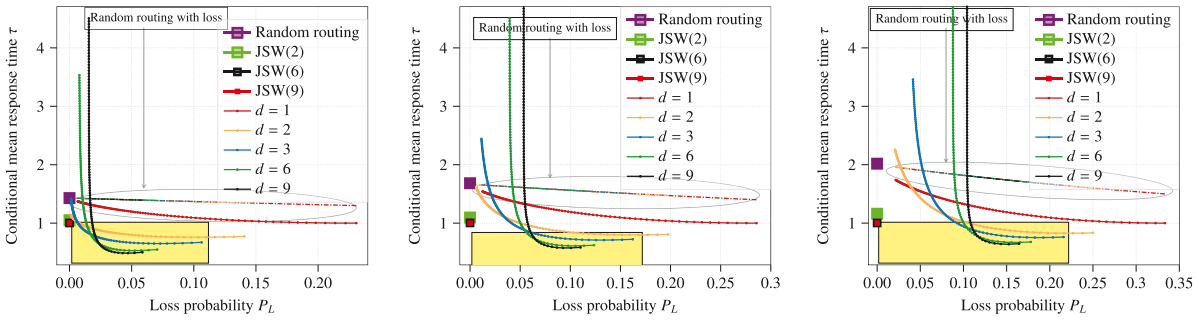
(a) τ vs threshold T. (b) P_L vs threshold T. (c) τ vs P_L tradeoff.

Fig. 3. We plot the conditional mean response time τ and the loss probability P_L for the policy $\pi(d,T,T)$ as a function of discard threshold T, for the number of servers $N = 20$, arrival rate $\lambda = 0.3$, service rate $\mu = 1$, and for the number of replicas $d \in \{1,2,3,6,9\}$.

In Fig. 2, we plot the behavior of conditional mean response time τ and the loss probability P_L for $\pi(d,T,T)$ as the number of replicas d increases. We choose the number of servers $N = 20$ and discard threshold $T = 1.5$. Such a study is relevant for determining the ideal choice for the number of replicas d for a given arrival rate. Here are the main observations.

1. Fig. 2(a) shows that there is an optimal number of replicas d that minimizes the conditional mean response time for each arrival rate. In addition, the optimal number of replicas d decreases with an increase in arrival rate. This is expected as when the system load increases with a finite value for both thresholds, the chance of replicas getting canceled increases. Even though, increasing the number of replicas ensures that more copies of the job are serviced in parallel, it results in an increase in load at individual servers. Thus, beyond a certain threshold, it can result in an increase in conditional response time. For reference, we have also plotted the mean response time for the random routing policy for the different arrival rates in Fig. 2(a).
2. Fig. 2(b) demonstrates that there is again an optimal number of replicas d which minimizes the loss probability for each arrival rate. For a small number of replicas, there is a high probability of the job getting canceled, since we are sampling fewer servers. However, a larger number of replicas d can cause an increase in workload at the servers, which again results in an increase in the cancellation of replicas. Server workloads increase with the arrival rate, and hence the loss probability increases with the arrival rate.
3. From the tradeoff presented in Fig. 2(c), it is clear that we can determine a suitable replication factor d that minimizes both the conditional mean response time and the loss probability simultaneously for each value of arrival rate. Further, this optimal number of replicas decreases with an increase in arrival rate.

In Fig. 3, we plot the behavior of the conditional mean response time τ and the loss probability P_L for the $\pi(d,T,T)$ policy as a function of the discard threshold T. We choose the number of servers $N = 20$, the normalized arrival rate $\lambda = 0.3$, and the number of replicas $d \in \{1,3,6,9\}$. In addition, we compare the performance of the proposed policy with loss probability P_L to a lossy random routing random dropping policy, where there is a Poisson arrival of $N\lambda$ to a system of N independent servers with *i.i.d.* exponential service times of rate μ. We assume that each arrival is dropped with probability P_L, and a non-dropped arrival is routed to one of the N servers uniformly at random. The mean response time for the random routing random dropping policy is $\frac{\mu}{\mu-\lambda(1-P_L)}$. We list our main observations below.



(a) τ vs P_L tradeoff, $\lambda = 0.3$. (b) τ vs P_L tradeoff, $\lambda = 0.4$. (c) τ vs P_L tradeoff, $\lambda = 0.5$.

Fig. 4. For each arrival rate $\lambda \in \{0.3, 0.4, 0.5\}$, we plot the tradeoff between conditional mean response time τ and the loss probability P_L for the policy $\pi(d, T, T)$ as a function of discard threshold T, for the number of servers $N = 20$, service rate $\mu = 1$, and for the number of replicas $d \in \{1, 2, 3, 6, 9\}$.

1. From Fig. 3(a), we see that the discard threshold T that minimizes the conditional mean response time varies with the choice of replication factor d. Since fewer replicas will be discarded as the threshold T increases, we expect the loss probability P_L to decrease with the discard threshold T. We verify this behavior in Fig. 3(b).
2. When the discard threshold $T \in [0, 1]$, we see significant reduction in conditional mean response time under $\pi(d, T, T)$ when compared to random routing. This gain comes at the cost of a nominal loss probability P_L for $d \geqslant 3$. In fact, the maximum loss probability is observed to be around 0.095 for $d = 3$.
3. The tradeoff curve in Fig. 3(c) helps in determining the best discard threshold T for a fixed replication factor d. It suggests that with an increase in the number of replicas, decreasing the discard threshold could be beneficial as the corresponding increase in loss probabilities is nominal. It also provides a comparison with the conditional mean response time for the JSW(d) policy. For the considered arrival rate, it shows that the proposed policy beats the JSW(d) policy if a loss is allowed and this loss to be admitted increases with d. To be specific, the loss percentage to be borne while using the proposed policy in order to provide a better performance than the JSW(d) policy are $0.65, 1.35$, and 1.8 when the number of replicas is 3, 6, and 9 respectively for a normalized arrival rate of 0.3.
4. We also compare the performance of our policy with the random routing random dropping policy described earlier. From Figs. 3(a) and 3(c), we can see that $\pi(d, T, T)$ outperforms this policy while maintaining the same loss probability.

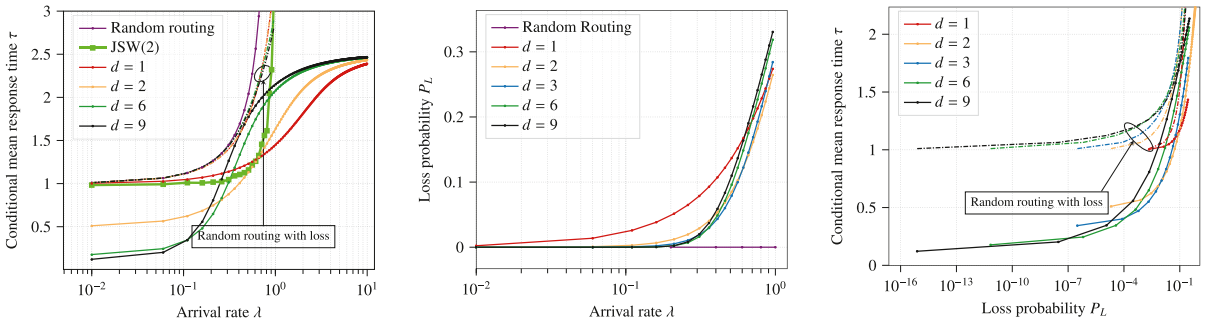
Fig. 4 presents similar plots as Fig. 3(c) but for different normalized arrival rates. They show that the loss probability to be admitted by the proposed policy in order to provide a competitive performance to that of the JSW(d) policy increases with the increase in arrival rate. In Fig. 5, we study the behavior of conditional mean response time τ and the loss probability P_L for $\pi(d, T, T)$ as the normalized arrival rate λ increases. We choose the number of servers $N = 20$, discard threshold $T = 1.5$, and the number of replicas $d \in \{1, 3, 6, 9\}$. We list our observations and inferences below which are similar to other discard thresholds.

1. Fig. 5(a) shows that the conditional mean response time for $\pi(d, T, T)$ policy for $d > 1$ is uniformly smaller than random routing for all arrival rates. These performance improvements come at the cost of some nominal loss probability for low arrival rates.
2. Since the $\pi(d, T, T)$ policy admits loss, we observe from Fig. 5(a) that the conditional response time remains bounded even for higher arrival rates. However, this property results in a non-trivial loss probability for higher arrival rates, as seen in Fig. 5(b).
3. From the tradeoff in Fig. 5(c), we again infer that as the arrival rate increases, it is wiser to switch to a lower number of replicas.
4. As observed earlier, $\pi(d, T, T)$ can be seen to outperform the random routing with random dropping policy (see Figs. 5(a) and 5(c)).

Remark 10. As mentioned earlier, this policy is to be adopted only in applications that can tolerate a certain amount of loss such as streaming applications. In addition, the optimal value of policy parameters d and T depends on the application, especially on the minimum tolerable loss probability for the given application. We also note that joint optimization of these parameters is difficult to perform analytically. However, in practice, one can always use the derived expressions to find the best operating point through approaches such as grid search.

4.2. Replication with no loss

We next study the N server system under the replication with no loss policy. Specifically, we assume that the primary discard threshold $T_1 = \infty$, and the secondary discard threshold $T_2 < T_1$ is finite. In this case, the system is stable if and only if $\lambda < \mu$. First, we obtain the following result from Corollary 10 by substituting $T_1 = \infty$.



(a) τ vs λ. (b) Loss probability P_L vs λ. (c) τ vs P_L tradeoff.

Fig. 5. We plot the conditional mean response time τ and the loss probability P_L as the normalized arrival rate λ increases under policy $\pi(d, T, T)$ for the number of servers $N = 20$, discard threshold $T = 1.5$, service rate $\mu = 1$, and the number of replicas $d \in \{1, 2, 3, 6, 9\}$.

Corollary 15. *For an N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, the stationary workload distribution at the cavity queue under $\pi(d, \infty, T_2)$ policy exists only for $\lambda < \mu$, and is given by*

$$F(w) = \begin{cases} F(0)\left(\frac{\mu}{\mu-\bar{\lambda}} - \frac{\bar{\lambda}}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})w}\right), & w \leqslant T_2 \\ F(T_2) + \frac{\bar{\lambda}}{\mu-\lambda} F(0) e^{(\bar{\lambda}-\lambda)T_2}\left(e^{-(\mu-\lambda)T_2} - e^{-(\mu-\lambda)w}\right), & w > T_2, \end{cases}$$

where $F(0) = \frac{(1-\frac{\lambda}{\mu})(1-\frac{\bar{\lambda}}{\mu})}{(1-\frac{\lambda}{\mu})+\frac{\bar{\lambda}}{\mu}(\frac{\lambda}{\mu}-\frac{\bar{\lambda}}{\mu})e^{-(\mu-\bar{\lambda})T_2}}$.

Remark 11. Note that the loss probability is 0 under this policy. Then, from Theorem 7, we have

$$\tau = \int_x \left[k(x, \infty)(\bar{F}(T_2) + k(x, T_2))^{d-1}\right] dx. \tag{8}$$

The next lemma provides us with the tail distributions $k(x, T_2)$, $k(x, \infty)$, and $\bar{F}(T_2)$ that enable us to compute the stationary mean response time τ under the scheduling policy $\pi(d, \infty, T_2)$. Note that, we provide the results only for the regime of arrival rates where the system is stable, that is when $\lambda < \mu$.

Lemma 16. *For a stable N server system with i.i.d. exponential service times of rate μ and Poisson arrivals of rate $N\lambda$, the stationary tail distribution of sojourn time $k(x, T_2)$ for an undiscarded job under the $\pi(d, \infty, T_2)$ policy is*

$$k(x, T_2) = \begin{cases} F(0) e^{-\mu x} e^{\bar{\lambda} T_2}, & x \geqslant T_2 \\ F(0)\left[\frac{\mu}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})x} - \frac{\bar{\lambda}}{\mu-\bar{\lambda}} e^{-(\mu-\bar{\lambda})T_2}\right], & x < T_2. \end{cases}$$

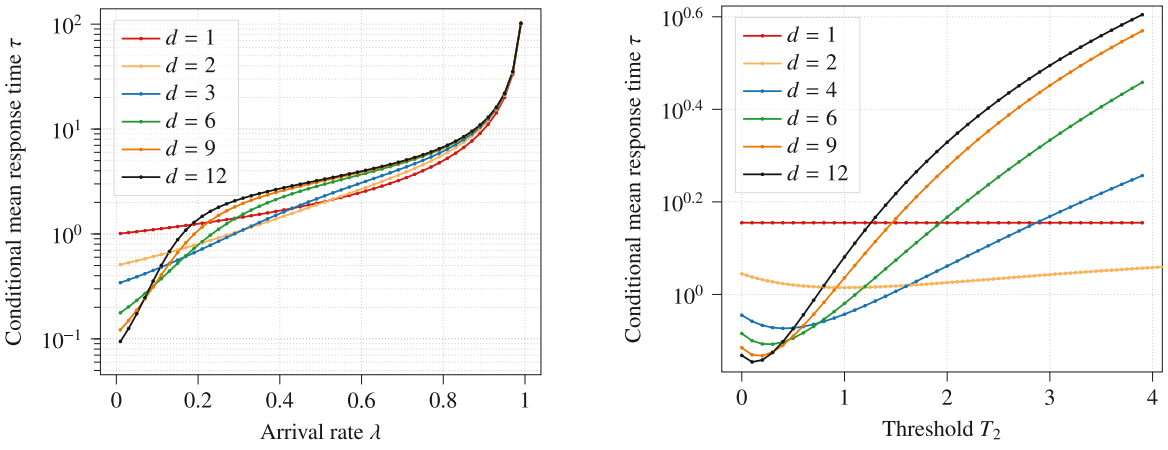
The probability mass $F(0) = \left[\bar{\lambda}\left(\frac{1-e^{-(\mu-\bar{\lambda})T_2}}{\mu-\bar{\lambda}} + \frac{e^{-(\mu-\bar{\lambda})T_2}}{\mu-\lambda}\right) + 1\right]^{-1}$, *and we can find the limit $k(x, \infty)$ as*

$$k(x, \infty) = k(x, T_2) + \begin{cases} F(0)\bar{\lambda} e^{(\bar{\lambda}-\lambda)T_2} e^{-\mu x}\left[\frac{e^{\lambda x} - e^{\lambda T_2}}{\lambda} + \frac{e^{\lambda x}}{\mu-\lambda}\right], & x \geqslant T_2 \\ \frac{\bar{\lambda}}{\mu-\lambda} F(0) e^{-(\mu-\bar{\lambda})T_2}, & x < T_2. \end{cases}$$

Proof. Since the service time is exponentially distributed with rate μ, we get $\bar{G}(x) = e^{-\mu(x)_+}$. Therefore, we can write the function $k(x, T) = \mathbb{E}\left[\bar{G}(x - W)\mathbb{1}_{\{W \leqslant T\}}\right] = F(T) - F(T \wedge x) + e^{-\mu x}\int_0^{T \wedge x} e^{\mu w} dF(w)$. Setting $T = \infty$ in the above equation, we get $k(x, \infty) = F(\infty) - F(x) + e^{-\mu x}\int_0^x e^{\mu w} dF(w)$. Substituting the workload distribution F from Corollary 15, we get the result. □

We compare the mean response time τ for jobs under policy $\pi(d, \infty, T_2)$ in Fig. 6 for different number of replicas d, when the number of servers $N = 20$ and the exponential service rates of jobs is $\mu = 1$. We plot τ as a function of normalized arrival rate λ in Fig. 6(a), where we select a secondary discard threshold $T_2 = 2$ which is twice the mean service time of a job. We plot τ as a function of secondary discard threshold T_2 in Fig. 6(b), where we choose the normalized arrival rate $\lambda = 0.3$. We list out the observations in the following.

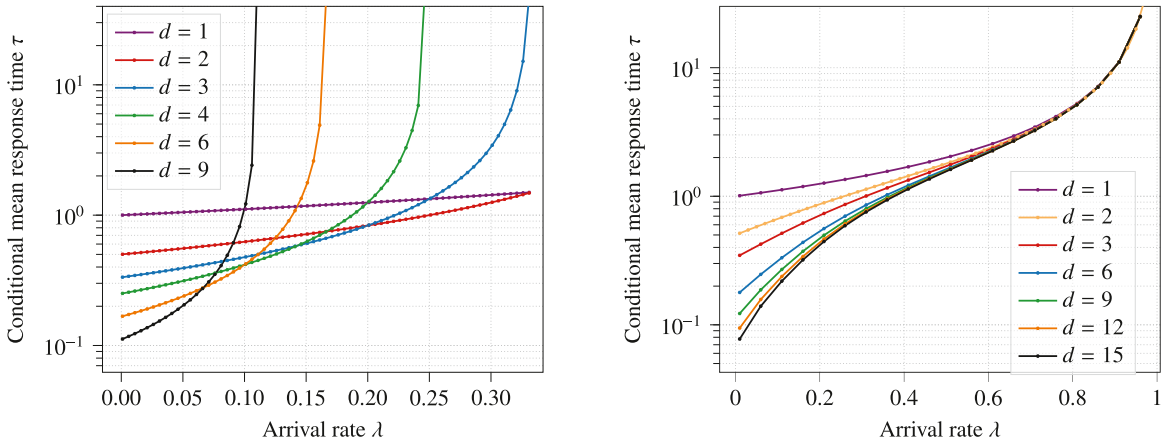
1. Fig. 6(a) shows that the lower replication factor d is preferable for larger arrival rates λ. This is due to the fact that system load increases due to a larger number of redundant replicas, adversely impacting the mean response time performance at high arrival rates.



(a) τ vs arrival rate λ for discard threshold $T_2 = 2$.

(b) τ vs discard threshold T_2 for arrival rate $\lambda = 0.3$

Fig. 6. We plot the mean response time τ for the policy $\pi(d, \infty, T_2)$ for the number of servers $N = 20$, service rate $\mu = 1$, and the number of replicas d.



(a) τ vs arrival rate λ for discard threshold $T_2 = \infty$.

(b) τ vs arrival rate λ for discard threshold $T_2 = 0$.

Fig. 7. We plot the mean response time τ as a function of arrival rate λ for $\pi(d, \infty, T_2)$ policy for a fixed number of servers $N = 20$, service rate $\mu = 1$, and number of replicas $d \in \{1, 2, 3, 6, 9\}$.

2. Fig. 6(b) shows the existence of an optimal discard threshold T_2 for a fixed number of replicas d, and this optimal threshold decreases with an increase in the number of replicas.

To conclude, as the normalized arrival rate increases, it is preferable to decrease the number of replicas while choosing an appropriate value for the secondary discard threshold.

Remark 12. The $\pi(d, \infty, \infty)$ policy is a special case of $\pi(d, T, T)$ for $T = \infty$ as well as of $\pi(d, \infty, T_2)$ for $T_2 = \infty$. We note that no jobs are lost in such a system and therefore, the loss probability is zero. This is an r.w.c. policy and has been studied in [55]. Under this policy, the arrival rate to any queue is $\bar{\lambda}$, and hence the system is stable if only if $\bar{\lambda} < \mu$. Using Lemma 13 it can be shown that $k(x, \infty) = e^{-(\mu - \bar{\lambda})x}$ for this policy under stability. Using this, the mean response time for exponential service time distribution can be found to be $\tau = \frac{1}{(\mu - \bar{\lambda})d}$.

We plot the mean response time τ for policy $\pi(d, \infty, \infty)$ as a function of arrival rate λ in Fig. 7(a), for the number of servers $N = 20$, service rate $\mu = 1$, and different number of replicas d. The figure is indicative of the stability condition $\lambda < \frac{1}{d}$ for this policy. The performance gain from using larger values of d is also evident, but this comes at the cost of requiring a stricter stability condition. Of course, the clear advantage of this policy over random routing ($d = 1$) is limited to lower arrival rates. At higher arrival rates λ, the fact that the redundant replicas cannot be canceled adversely impacts the system performance. For better clarity, we also provide the percentage improvement of the mean response time of the policy $\pi(d, \infty, \infty)$ over random routing policy *across stable regions* in Table 1.

Table 1
Percentage improvement in the mean response time of the policy $\pi(d,\infty,\infty)$ over random routing with the number of servers $N = 20$ and service rate $\mu = 1$. See Fig. 7(a).

Replicas	$\lambda = 0.1$	$\lambda = 0.15$	$\lambda = 0.2$	$\lambda = 0.25$
d = 2	43.6%	39.18%	33.19%	24.79%
d = 3	57%	48.26%	32.91%	−1%
d = 4	62.29%	46.4%	−1.91%	NA

From the above studies, we observe that the introduction of secondary replicas adds to the system load and deteriorates the system performance for high arrival rates. Therefore, in the following section, we study a policy where secondary replications occur only on idle servers.

4.3. Replication on idle secondary servers

As mentioned above, we next study the special case of $\pi(d,\infty,T_2)$ policy where the secondary discard threshold $T_2 = 0$. In this case, the secondary replicas are added only if the sampled secondary servers are idle. Here, we would like to point out a seemingly similar policy which is the Redundant to idle queue (RIQ(d)) [53]. We note that, unlike our policy which utilizes absolutely no feedback information, the RIQ(d) policy utilizes information about the availability of idle servers. If there is no more than a single idle server, RIQ(d) policy is identical to the Join Threshold Queue (JTQ(d,T)) [22, Section 6.6] with threshold T set to zero which we discuss in Section 5. Although the RIQ(d) policy is studied under a more general service model, the analysis is only approximate and no closed-form expressions are provided for the performance metrics under this general model. On the other hand, we provide closed-form expressions for mean workload under our proposed policy with *i.i.d.* exponential service times. In addition, we have also provided implicit expressions for general *i.i.d.* service times.

The replication on idle secondary servers policy that we discuss here is a special case of replication with no loss policy and we can obtain the limiting mean response time directly from the previously obtained result.

Lemma 17. *The limiting mean response time of any job under the dispatching policy $\pi(d,\infty,0)$ when service times of each job is i.i.d. exponential with rate μ and arrivals are Poisson with rate $N\lambda$, is given by*

$$\tau = \sum_{n=0}^{d-1} \binom{d-1}{n} \bar{F}(0)^{d-1-n} F(0)^{n+1} \left[\frac{d\mu\lambda}{(\mu-\lambda)(\mu(n+1)-\lambda)\lambda} - \frac{\lambda(d-1)}{\lambda\mu(n+1)} \right], \tag{9}$$

for $\lambda < \mu$, $F(0) = \frac{\mu-\lambda}{\mu+\lambda(d-1)}$, and $\bar{F}(0) = 1 - F(0)$.

Proof. Substituting $T_2 = 0$ in Lemma 16 and substituting the terms in (8) gives the result. □

Remark 13. To better understand the behavior of $\pi(d,\infty,0)$ policy, we can simplify the expression for tail response time distribution for the cavity queue under this policy with $N\lambda$ Poisson arrivals and *i.i.d.* exponential service rate 1 as $\bar{H}_d(x) = k(x,\infty)(\bar{F}(0) + k(x,0))^{d-1} = e^{-x}\left(1 + \frac{d(e^{\lambda x}-1)}{(d-1)\lambda+1}\right)\left(1 - \frac{(1-\lambda)(1-e^{-x})}{(d-1)\lambda+1}\right)^{d-1}$.

The next Lemma shows that the response time under $\pi(d,\infty,0)$ policy is stochastically decreasing in d. Since random routing corresponds to $\pi(d,\infty,0)$ policy for $d = 1$, this Lemma implies that the performance of the $\pi(d,\infty,0)$ policy can never be worse than that of random routing.

Lemma 18. *The response time under the policy $\pi(d,\infty,0)$ with $N\lambda$ Poisson arrivals and i.i.d. exponential service rate 1 is stochastically decreasing in d.*

Proof. In order to show the stochastic ordering, it suffices to show that the tail response time follows $\bar{H}_{d+1}(x) \leqslant \bar{H}_d(x)$ for all $x \in \mathbb{R}_+$ and $d \in \mathbb{N}$. To this end, we first observe from Remark 13 that $\bar{H}_d(x) = e^{-x}e^{f(d,x)}$ where the function $f : [1,\infty)\times\mathbb{R}_+ \to \mathbb{R}$ can be defined for each $y \geqslant 1$ and $x \in \mathbb{R}_+$ as

$$f(y,x) \triangleq (y-1)\ln(y\lambda + (1-\lambda)e^{-x}) + \ln(ye^{\lambda x} - (y-1)(1-\lambda)) - y\ln(y\lambda + 1 - \lambda). \tag{10}$$

We will show that $f(y,x)$ is nonincreasing in $y \in [1,\infty)$ for all $x \in \mathbb{R}_+$, and hence the result follows. It suffices to show that the first partial derivative of f with respect to y is upper bounded by zero. To this end, we write

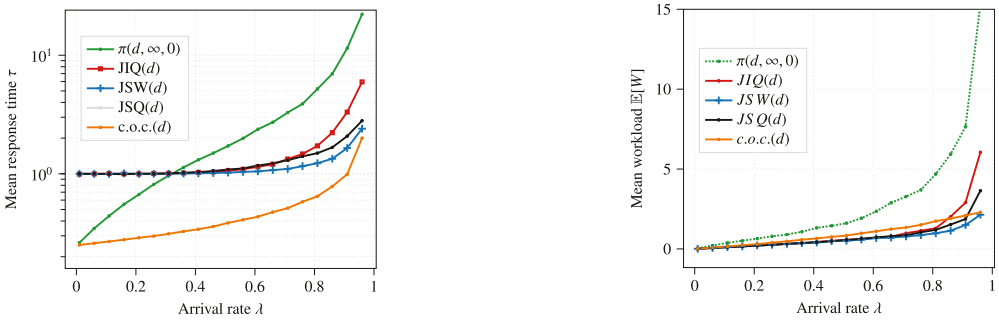
$$\frac{\partial f(y,x)}{\partial y} = \frac{e^{\lambda x} - 1 + \lambda}{y(e^{\lambda x} - 1 + \lambda) + 1 - \lambda} + \frac{(y-1)\lambda}{y\lambda + (1-\lambda)e^{-x}} - \frac{y\lambda}{y\lambda + 1 - \lambda} + \ln\left(1 - \frac{(1-\lambda)(1-e^{-x})}{y\lambda + 1 - \lambda}\right).$$

We recall that $(e^{\lambda x} - 1) \leqslant \lambda(e^x - 1)$ for all $\lambda \in [0,1]$ and $x \in \mathbb{R}_+$, $\frac{a}{ya+1-\lambda} \leqslant \frac{a_0}{ya_0+1-\lambda}$ for $0 < a \leqslant a_0$, and $\ln(1-x) \leqslant -x$ for all $x \in [0,1]$, to upper bound the partial derivative of f with respect to y as

$$\frac{\partial f(y,x)}{\partial y} \leqslant \frac{y\lambda}{y\lambda + (1-\lambda)e^{-x}} - \frac{y\lambda + (1-\lambda)(1-e^{-x})}{y\lambda + 1 - \lambda} = -\frac{(1-\lambda)^2 e^{-x}(1-e^{-x})}{(y\lambda + 1 - \lambda)(y\lambda + (1-\lambda)e^{-x})} \leqslant 0. \quad \square$$

Table 2
Percentage improvement in mean response time of the policy $\pi(d, \infty, 0)$ over random routing with the number of servers $N = 20$ and service rate $\mu = 1$. See Fig. 7(b).

Replicas d	$\lambda = 0.2$	$\lambda = 0.4$	$\lambda = 0.6$	$\lambda = 0.8$
3	43.14%	22.02%	7.9%	1.74%
6	57.23%	29.30%	10.37%	2.22%
9	62.33%	31.97%	11.22%	2.39%
12	64.96%	33.35%	11.66%	2.47%



(a) τ vs λ (b) $\mathbb{E}[W]$ vs λ

Fig. 8. The mean response time τ and the mean workload $\mathbb{E}[W]$ at the cavity queue as a function of normalized arrival rate λ for the policies $\pi(d, \infty, 0)$, JIQ(d), JSW(d), JSQ(d), c.o.c.(d) for exponential service distribution of rate $\mu = 1$, the number of servers $N = 20$ and the number of replicas $d = 4$.

When the system is lightly loaded, we expect that the replicas of an arriving job will find most servers idle. Thus, all d replicas get served by d parallel servers, leading to improvement in the mean response time performance. However, under heavy traffic regimes, this policy behaves similarly to the random routing policy where only the primary replica gets served, while all secondary replicas are likely to get canceled. In this regime, the policies with queue state information can perform better, although this improvement in performance comes at the cost of procurement of information. We plot the mean response time τ for policy $\pi(d, \infty, 0)$ as a function of arrival rate λ in Fig. 7(b), for the number of servers $N = 20$, service rate $\mu = 1$, and different number of replicas d. Here, are the main observations.

1. The mean response time for $\pi(d, \infty, 0)$ is uniformly better for larger number of replicas d, and the gains are highest for lower arrival rates.
2. Here, the additional replicas are executed only if the server is idle in this policy. Therefore, a higher choice of replication factor d does not increase the system load significantly.
3. For moderate to higher values of arrival rates, all the different choices of the number of replicas have a similar performance under the stability region of $\lambda < \mu$, independent of the number of replicas d.

We also provide the percentage improvement of the conditional mean response time of the policy $\pi(d, \infty, 0)$ over random routing policy for various values of normalized arrival rate in Table 2.

We note from the numerical comparisons that $\pi(d, \infty, 0)$ policy offers a superior performance among all $\pi(d, \infty, T_2)$ policies. It is also clear that $\pi(d, \infty, 0)$ policy performs better than random routing for any value of d. We now proceed to study the performance of this policy with respect to some of the best-known load-balancing policies in the literature.

5. Comparison with feedback-based policies

As a benchmark, we compare the performance of the proposed $\pi(d, \infty, 0)$ policy to some popular policies like redundancy-d cancel on start (c.o.s.), redundancy-d cancel on complete (c.o.c.), JSQ(d), and Join Threshold Queue (JTQ(d, T)) [22, Section 6.6] that have information feedback and/or synchronized cancellation of replicas. We choose to set threshold T to 0 in the JTQ(d, T) policy where it is identical to the JIQ(d) policy. It is easy to see that the c.o.s.(d) policy is identical to the JSW(d) policy. For fairness of comparison, we are comparing only no-loss policies, in which case, the conditional mean response time is the mean response time for any job. We first present analytical comparison between c.o.s.(d) and c.o.c.(d) followed by the comparison between c.o.c.(d) and the proposed $\pi(d, \infty, 0)$ policy. Then, we proceed to present the comparison through simulation studies. The unavailability of closed-form expressions for some of the existing policies under comparison and the complicated expressions for the mean response time distributions restrict us from providing a complete analytical comparison between all the policies considered.

Remark 14. For a large N server system with a Poisson arrival rate of $N\lambda$, *i.i.d.* exponential service times of mean 1, assuming asymptotic independence of marginal workloads at stationarity, the limiting tail response time distribution for all $x \in \mathbb{R}_+$, under

replication-d for $d \geqslant 2$ with c.o.s. [4, Theorem 5.3] and c.o.c. [18, Theorem 6], is

$$\bar{H}_{\text{c.o.s.}(d)}(x) = \frac{1}{(\lambda^d + (1-\lambda^d)e^{(d-1)x})^{\frac{1}{d-1}}}, \quad \bar{H}_{\text{c.o.c.}(d)}(x) = \frac{1}{(\lambda + e^{(d-1)x}(1-\lambda))^{\frac{d}{d-1}}}.$$

We next show that cancel-on-complete(d) always outperforms cancel-on-start(d) and $\pi(d, \infty, 0)$ policy for *i.i.d.* exponential service. In particular, we show that $\bar{H}_{\text{c.o.c.}(d)}(x) \leqslant \bar{H}_{\text{c.o.s.}(d)}(x)$ and $\bar{H}_{\text{c.o.c.}(d)}(x) \leqslant \bar{H}_{\pi(d,\infty,0)}(x)$ for all $x \in \mathbb{R}_+$ and number of replicas $d \geqslant 2$.

Proposition 19. *For an N server system with a Poisson arrival rate of $N\lambda$, i.i.d. exponential service times of mean 1, the stationary response time of c.o.c.(d) policy is always stochastically dominated by that of c.o.s.(d) policy. That is, $\bar{H}_{\text{c.o.c.}(d)}(x) \leqslant \bar{H}_{\text{c.o.s.}(d)}(x)$ for all $x \in \mathbb{R}_+$ and $d \geqslant 2$.*

Proof. Let $x \in \mathbb{R}_+$ and $d \geqslant 2$. From Remark 14, we observe that we only need to show $\lambda^d + (1-\lambda^d)e^{(d-1)x} \leqslant (\lambda + e^{(d-1)x}(1-\lambda))^d$. However, it follows from the observation $(\lambda + e^{(d-1)x}(1-\lambda))^d \geqslant \lambda^d + e^{d(d-1)x}(1-\lambda)^d \geqslant \lambda^d + e^{(d-1)x}(1-\lambda)^d$. □

Proposition 20. *For an N server system with a Poisson arrival rate of $N\lambda$, i.i.d. exponential service times of mean 1, the response time of c.o.c.(d) policy is stochastically dominated by that of $\pi(d, \infty, 0)$ policy. That is, $\bar{H}_{\text{c.o.c.}(d)}(x) \leqslant \bar{H}_d(x)$ for all $x \in \mathbb{R}_+$ and $d \geqslant 2$.*

Proof. From Remarks 13 and 14, it suffices to show that for all $x \in \mathbb{R}_+$

$$\frac{1}{(\lambda + e^{(d-1)x}(1-\lambda))^{\frac{d}{d-1}}} \leqslant e^{-x}\Big(1 + \frac{d(e^{\lambda x}-1)}{(d-1)\lambda + 1}\Big)\Big(1 - \frac{(1-\lambda)(1-e^{-x})}{(d-1)\lambda+1}\Big)^{d-1}.$$

In order to prove this, we consider a function $g : [2, \infty) \times \mathbb{R}_+ \to \mathbb{R}$ defined as

$$g(y,x) \triangleq -x + \ln(y(e^{\lambda x}-1) + y\lambda + 1 - \lambda) + (y-1)\ln(y\lambda + (1-\lambda)e^{-x}) - y\ln(y\lambda + 1 - \lambda) + \frac{y}{y-1}\ln(\lambda + e^{(y-1)x}(1-\lambda)).$$

We observe that $g(y, 0) = 0$. Then, we obtain the required result by showing that $g(y, x)$ is increasing in x for all $x \in \mathbb{R}_+$. To this end, we compute the first partial derivative of $g(y, x)$ with respect to x, and write

$$\frac{\partial g(y,x)}{\partial x} = -(1-\lambda) + \frac{\lambda(1-\lambda)(y-1)}{y(e^{\lambda x}-1) + y\lambda + 1 - \lambda} - \frac{(y-1)(1-\lambda)e^{-x}}{y\lambda + (1-\lambda)e^{-x}} + \frac{y(1-\lambda)e^{(y-1)x}}{\lambda + (1-\lambda)e^{(y-1)x}}.$$

We use the fact that $e^{\lambda x} - 1 \leqslant \lambda(e^x - 1)$ for all $\lambda \in [0, 1]$ and $x \in \mathbb{R}_+$, to observe that $g(y, x)$ is increasing in x as

$$\frac{1}{(1-\lambda)}\frac{\partial g(y,x)}{\partial x} \geqslant (y-1) - \frac{(y-1)(1-\lambda)e^{-x}}{y\lambda + (1-\lambda)e^{-x}} + \frac{y\lambda(e^{(y-1)x}-1)}{\lambda + (1-\lambda)e^{(y-1)x}} = \frac{y(y-1)\lambda}{y\lambda + (1-\lambda)e^{-x}} + \frac{y\lambda(e^{(y-1)x}-1)}{\lambda + (1-\lambda)e^{(y-1)x}} \geqslant 0. \quad \square$$

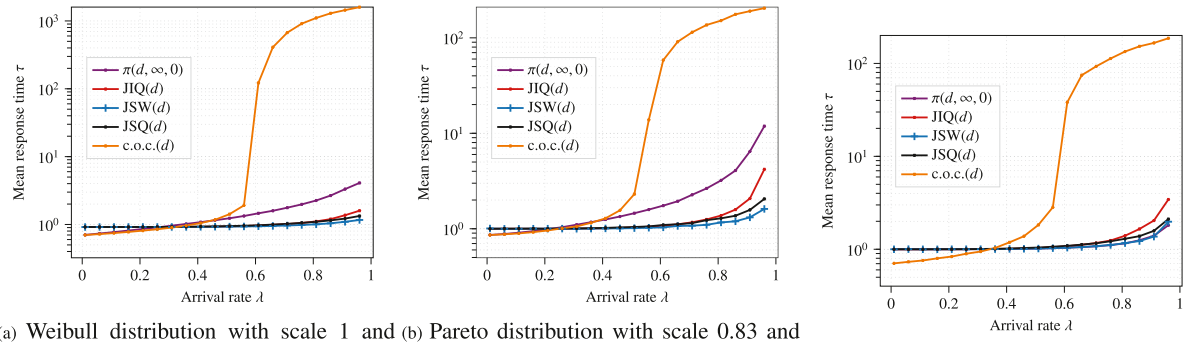
Although closed-form expressions for mean response time for c.o.s.(d) and c.o.c.(d) policies can be found in [4, Theorem 5.4] and [18, Theorem 6] respectively, we do not have closed-form expressions for the mean response time of JIQ(d) and JSQ(d) policies under the given settings. The comparison of these policies against the proposed policy via numerical simulations is presented next. All the experiments reported in this section have been run for 10^5 iterations with the number of servers $N = 20$ and the number of replicas $d = 4$.

In Fig. 8(a), we plot the mean response time τ for JIQ(d), JSW(d), JSQ(d) and c.o.c.(d) policies against the replicate on idle secondary servers ($\pi(d, \infty, 0)$) policy when the service times are *i.i.d.* exponentially distributed with rate 1. From the figure, we observe that the mean response time for the $\pi(d, \infty, 0)$ policy for low arrival rates is lower than that of JIQ(d), JSW(d), and JSQ(d) policies. Although at higher arrival rates, the JIQ(d), JSW(d), and JSQ(d) policies perform better than our policy, this performance improvement comes at the price of information exchange between the servers and dispatcher. In addition, we observe that the cancel-on-complete policy performs the best for all arrival rates. This is due to the fact that cancel on complete is equivalent to water filling at the d sampled servers for *i.i.d.* exponential service [45], and the water filling policy has an additional degree of freedom to divide jobs arbitrarily on different servers. It should also be kept in mind that c.o.c. policy requires strict coordination and communication among the servers to achieve this performance. Moreover, we will see in the next section the performance improvement of c.o.c. policy degrades for non-exponential service distributions like Weibull and Pareto and it further suffers from stability issues.

We now provide a comparison of the expected workloads at the cavity queues in each of these policies in Fig. 8(b). Compared to other policies, the expected workload at the individual queues is higher in our policy. This is expected as a larger number of replicas are processed per job under our policy, unlike the other policies that perform coordinated cancellation of additional replicas. However, the extra workload is not huge in the low arrival rate regime. Also, to be noted is that our policy provides a performance improvement in this regime in spite of the increment in the average workload. In fact, the increment of the workload is caused by the additional redundant replicas and it is this additional redundancy that helps in bringing down the overall response time of the job.

5.1. General service time distribution

We see from our previous analysis that obtaining closed-form expressions for the mean response time for our policy can be difficult when service times are not exponentially distributed. In this section, we provide observations on numerical studies



(a) Weibull distribution with scale 1 and shape 5.

(b) Pareto distribution with scale 0.83 and shape 5.5.

(c) Uniform distribution in [0.5, 1.5].

Fig. 9. The mean response time as a function of normalized arrival rate λ for the policies JIQ(d), JSW(d), JSQ(d), c.o.c.(d) with respect to the $\pi(d, \infty, 0)$ policy, for the number of servers $N = 20$ and the number of replicas $d = 4$.

conducted on our policy under non-exponential service time distributions. In [22], the authors discuss the analysis for several workload-dependent load-balancing policies when job sizes follow a general distribution. However, closed-form expressions are lacking and the solution is determined numerically. Further, the authors of [56] provide a method to derive the expected workload at the cavity queue of the N server system when the jobs are serviced only when the workload at arrival is less than a threshold and when service times are *i.i.d.* and follow a general distribution. They provide expressions, implicit in some cases, for the expected workload when the service times are deterministic or follow phase type, Erlang, or exponential distribution. This setting matches our special case of $\pi(d, T, T)$ policy and their expressions hold valid for this special case. However, the computation of mean response time requires numerical evaluations. Therefore, we do not adopt this methodology in our work and we provide only simulation results for the comparison of our policy under non-exponential service time distributions.

The mean response time τ of JIQ(d), JSW(d), JSQ(d) and c.o.c.(d) policies against $\pi(d, \infty, 0)$ policy is plotted as a function of normalized arrival rate λ in Figs. 9(a)–9(c) when the service times follow Weibull distribution with scale parameter 1 and shape parameter 5, Pareto distribution with scale parameter 0.83 and shape parameter 5.5, and uniform distribution in the range $[0.5, 1.5]$, respectively. We observe that except for the c.o.c.(d) policy, the behavior of the remaining policies remains similar to that in Fig. 8(a) for the exponential distribution case. We observe that the response time performance of c.o.c.(d) policy degrades and the stability region shrinks with the change in service time distribution. The $\pi(d, \infty, 0)$ policy achieves almost the same performance as c.o.c. without any coordination or communication requirements in the low to medium arrival rate regimes under Weibull and Pareto distributions. From moderate to high arrival rate regime, the c.o.c. policy tends to get unstable and the proposed policy is superior to c.o.c.(d) in this regime for non-exponential service distributions. This plot also demonstrates that the performance improvement of the proposed policy against feedback-based policies is not an artifact of choosing exponential service times.

6. Discussion & future work

In this work, we consider load-balancing policies without feedback and propose a policy based on timed replicas. For every replica that is created, the policy sends cancellation instructions to servers along with the replica. This instruction specifies an expiry time for the replica and thereby prevents potentially wasteful replicas from being executed. In this work, we have shown that this policy and several of its special cases, offer a marked improvement over the random routing policy for a suitable choice of parameters such as normalized arrival rate λ and number of replicas d. We also observed that under certain parameter regimes, the proposed dispatch policy has better performance when compared to feedback-based policies. We analyze this policy using the cavity queue approach and the assumption of asymptotic independence of queues at stationarity. Using the MGF approach, we characterize the limiting mean conditional response time of an undiscarded job and the limiting loss probability for the proposed policy. Our results partially address the two questions we raised in the introduction.

1. We have proposed a load-balancing policy without any server feedback that outperforms existing policies with the server feedback in certain operating regimes.
2. The existing load balancing policies with server feedback information are not optimally utilizing the feedback information, since there exists a no-feedback policy that outperforms them in certain operating regimes.

A key assumption in most of our analysis has been the exponential service requirements for jobs, and that the job replicas require *i.i.d.* service time. We believe that relaxing these assumptions and analyzing the proposed $\pi(d, T_1, T_2)$ policy for more general service time distributions and for the case of identical replicas is an interesting open direction. One can think of more nuanced policies. One such policy that does not require any server feedback, would replicate only short jobs if the service requirement of a job is known at arrival. One can also think of incorporating feedback in our proposed policy, and consider replicating only if the primary copy is discarded, or decide the number of replicas based on the queue state. Further, while the performance of $\pi(d, T_1, T_2)$ is good for low values of normalized arrival rates λ, it would be interesting to investigate if there exist other zero feedback policies that

are better than feedback-based policies even for higher values of normalized arrival rates λ. Another interesting direction is to find load-balancing policies that minimize the limiting mean of response times, and policies that can utilize server feedback in a more efficient way. Finally, we plan to study the use case for such no-feedback policies in an (n, k) fork-join system, where a parallelizable job is distributed across n servers and is considered completed when a certain fraction of jobs are executed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A. Proof sketch for Proposition 5

This proof for asymptotic independence of server workload during any finite time horizon, is based on the results provided in [44, Section 7]. Let the N dimensional workload process in the N server system be denoted by $W^N(t) \in \mathbb{R}^N$ for $t \geqslant 0$ under the given load balancing policy. The workload at queue k at time t is denoted by $W^{N,(k)}(t)$. We assume that all queues start with zero workloads, and hence are mutually independent. Let M^N be the measure of the workloads over the Borel sets $\mathcal{B}(\mathbb{R}^N)$ over $\mathbb{R}^N$. Also, let $M^{N,N'}$ denote the projection of M^N onto the first N' queues.

Recall that the replica service times are assumed to be independent across the queues and the arrival at each queue follows a Poisson distribution with rate λd. At any arrival instant, the set of d servers selected by the servers is referred to as the selection set corresponding to the arrival. In the load balancing policy studied in [44], whether a newly arrived job is accepted at a certain queue or not depends on the workloads of the other servers in the selection set. However, in our policy whether a job gets accepted at a queue or not depends only on the current workload at that queue. That is, suppose that the latest arrival to queue n before time t happened at time t'.

Under the policy of threshold-based cancellation, whether the job joins the queue k at time t' or not depends only on whether $W^{N,(k)}(t')$ is less than the preset threshold or not. That is, $W^{N,(k)}(t)$ depends only on $W^{N,(k)}(t')$ and service time of the arriving job. Recall that we assumed the workloads at all queues to be independent initially. However, if the job gets accepted at more than one queue in the corresponding selection set, then the correlated arrival of jobs in these queues will make the workloads dependent.

Let us introduce a measure $M^{T,\infty,N'}$ over $\mathbb{R}^{N'}$ which is the N' fold product of $M^{T,\infty,1}$. We need to prove that over a finite time horizon, the joint workload measure of any N' queues converges to the *i.i.d.* measure $M^{T,\infty,N'}$ asymptotically in the number of queues. We consider the convergence of the measures in total variation distance. More precisely, we need to prove that as $N \to \infty$, $\lim_{N\to\infty} \sup_{A\in\mathcal{B}(\mathbb{R}^{N'})} \left|M^{T,N,N'}(A) - M^{T,\infty,N'}(A)\right| = 0$. Next, we outline the main steps in the proof.

Step 1: Construction of an influence process and the number of influencing servers.

Consider a reversed time process $I^{N,N'}(T-t)$ for $t \in [0,T]$ constructed as given next. We define $I^{N,N'}(T) \triangleq \{1, 2, \ldots, N'\}$. Now, if there is a potential arrival at time $T-t$ at a queue $n \in I^{N,N'}((T-t)^-)$, then $I_{N,N'}(T-t) \triangleq I_{N,N'}((T-t)^-) \cup S$ where S is the selection set of d servers selected by the new arrival. Note that the knowledge of service times and intersecting selection sets at each arrival instant can completely describe the workload process $W^N(t)$. We define the number of influencing servers at time $T-t$ for $t \in [0,T]$, as

$$C^{N,N'}(T-t) \triangleq \left|I^{N,N'}(T-t)\right|$$

Step 2: Coupling the number of influencing servers with a d-ary branching process.

Couple the number of influencing servers $C^{N,N'}(T-t)$ with a process $C^{\infty,N,N'}(T-t)$ constructed as follows. We first let $C^{\infty,N,N'}(T) = C^{N,N'}(T) = N'$ and recursively define the process at each arrival instant t as $C^{\infty,N,N'}(T-t) \triangleq C^{\infty,N,N'}((T-t)^-) + d - 1$ if there is an intersection between the influence process $I^{N,N'}((T-t)^-)$ and the set of d servers selected by the arrival at time t. Note that $C^{\infty,N,N'}(T-t) \geqslant C^{N,N'}((T-t))$ always. We also observe that $C^{N,N'}(T-t) = C^{\infty,N,N'}(T-t)$ for all $t \in [0,T]$ if the selection set of servers at all arrival instants intersect with either zero or one server in $I^{N,N'}(T)$ within the time $[0,T]$. We may therefore think of $C^{\infty,N,N'}(T-t)$ as the number of influencing servers in an alternate system with the maximum intersection of one server for selections sets with $I^{N,N'}(T-t)$ over the time horizon T.

As the arrival to each queue occurs according to a Poisson process with rate λd, the process $C^{\infty,N,N'}((T-t))$ is a λd branching process. It follows from [44, Proposition 7.2] that the process $C^{N,N'}(T)$ converges to $C^{\infty,N,N'}(T)$ in probability for large N. Note that, $C^{N,N'}(T)$ being equal to $C^{\infty,N,N'}(T)$ guarantees that the number of influencing servers $C^{N,N'}(T-t)$ will be equal to the process $C^{\infty,N,N'}(T-t)$ for all $t \in [0,T]$.

Step 3: Extension of the influence process to an infinite server system.

We extend the influence process $I^{N,N'}(T-t)$ to the process $I^{\infty,N,N'}(T-t)$ satisfying $C^{\infty,N,N'}(T) = \left|I^{\infty,N,N'}(T)\right|$. As hinted earlier, we will be constructing an influence process $I^{\infty,N,N'}(T-t)$ in an alternate system comprising infinitely many servers where there is an intersection of at most one server for $I^{\infty,N,N'}(T-t)$ with selection sets over the time horizon T. Therefore, the workloads at the N' servers will stay independent over the time horizon $[0,T]$ in this alternate system. This extended influence process is constructed the same way as we construct $I^{N,N'}(T-t)$ except for the following. If an arrival happens at time t and if $C^{\infty,N,N'}(T-t) \neq C^{N,N'}(T-t)$,

include $(C^{\infty,N,N'}(T-t) - C^{N,N'}(T-t))$ new servers to the set $I^{\infty,N,N'}((T-t)^-)$ besides the ones in the new selection set. The inclusion of new servers is always possible as we suppose the system has infinitely many servers. Observe that the branches of this extended influence process starting from each of the N' servers will not intersect each other at any point and remain independent of each other. That is, the construction does not allow any correlated arrivals to occur for this workload process, preserving the independence of the N' servers under consideration. That is, the measure corresponding to the workload process at the N' servers of this infinite server system with the given influence process denoted as $M^{T,\infty,N'}$ will be the N fold product of $M^{T,\infty,1}$ due to the independence of the queues. From [44, Lemma 7.2], it can be seen that $M^{T,\infty,N'}$ is always independent of N. Further, whenever $C^{N,N'}(T) = C^{\infty,N,N'}(T)$, the processes $I^{N,N'}(T-t)$ and $I^{\infty,N,N'}(T-t)$ are also equal. As seen in Step 2 that $C^{N,N'}(T)$ converges to $C^{\infty,N,N'}(T)$ in probability for large value of N, the result follows.

Remark 15. We remark that the above proof does not make any assumption on the distribution of the service time except that they are *i.i.d.* across jobs.

The extension of this independence across the servers to infinite time intervals requires the following monotonicity conditions to be satisfied and does not follow directly. Let us first define the fraction of servers with workload greater than w at time $t \geqslant 0$ as $x_w^N(t) \triangleq \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{W^{N,(i)}(t)>w\}}$. Then, [45, Lemma 3] shows that the workloads under JSW(d) and water filling(d) dispatch policies, satisfy the following monotonicity property.

Proposition 21. *Consider two versions of the process, $x^N(.)$ and $\hat{x}^N(.)$, such that $x^N(0) \leqslant \hat{x}^N(0)$ Then these processes can be coupled so that, with probability* 1, *$x^N(t) \leqslant \hat{x}^N(t)$ for all $t \geqslant 0$.*

This monotonicity property holds for our proposed dispatch and cancellation policy only for the special case where the thresholds T_1 and T_2 are infinity. The following simplified example shows that the proposed $\pi(d, T_1, T_2)$ policy need not always offer this monotonicity property.

Example 2. Consider two single server systems with processes $x_w^1(t) = \mathbb{1}_{\{W^1(t)>w\}}$ and $\hat{x}_w^1(t) = \mathbb{1}_{\{\hat{W}^1(t)>w\}}$ indicating the events that the workloads $W^1(t)$ and $\hat{W}^1(t)$ at the respective servers in these two systems exceed the value w time t. The arrivals to the system follow Poisson distribution with rate λ, and arrival is accepted at the server only if the current workload at the server is less than a threshold T. We further suppose that the server services any job at a unit rate. Assume $W^1(0) = 0$ and $\hat{W}^1(0) = T + t'$ where t' is a positive constant. That is, $x_w^1(0) = 0 < \hat{x}_w^1(0) = 1$ for all $w < T + t'$. Suppose the first arrival happens at time $t_1 = t' - h$ which brings in a job of size $c > T + t'$. As the workload in the second system exceeds the threshold t, the new job is accepted only in the first system and the workloads in the coupled systems will be respectively $W^1(t_1) = c$ and $\hat{W}^1(t_1) = T + h$. That is, $x_w^1(t_1) = 1 > \hat{x}_w^1(t_1) = 0$ for $T_1 + h < w \leqslant c$. This shows that the monotonicity property need not hold under threshold-based policies.

We remark that the lack of monotonicity does not necessarily imply that the asymptotic independence of limiting marginal server workloads does not hold, and our simulation studies suggest that the assumption of asymptotic independence remains valid for our selected choice of system parameters.

Appendix B. Model validation

In this section, we discuss the accuracy of our theoretical results and compare them with simulation experiments. We obtained the conditional mean sojourn time τ for undiscarded jobs and the probability of discard P_L under the proposed probabilistic redundancy policy $\pi(d, T_1, T_2)$, based on the conjecture of the asymptotic independence of the queues. The workload distribution for the cavity queue under policy $\pi(d, T_1, T_2)$ has a closed-form expression for exponentially distributed service time, and is provided in Corollary 10. The expression for the conditional mean sojourn time under policy $\pi(d, T_1, T_2)$ is complex, and hence we have omitted it. Instead, we restrict our validation results to three special cases: (a) deterministic d replicas with identical finite discard threshold $\pi(d, T, T)$, (b) deterministic d replicas with no discard $\pi(d, \infty, \infty)$, and (c) deterministic d replicas with secondary replicas only at idle servers $\pi(d, \infty, 0)$.

Findings of the simulation experiments under the policy $\pi(d, T, T)$ are reported in Fig. B.10. We note that this is a lossy system, where some jobs can be discarded if none of the sampled servers have a workload smaller than the threshold T. We plot the conditional response time for $\pi(d, T, T)$ as a function of normalized arrival rate λ, when the jobs have *i.i.d.* exponential service times with unit mean. The identical discard threshold for primary and secondary replicas is taken as $T_1 = T_2 = 5$ and the total number of replicas is selected as $d = 3$. Each experiment is run over 10^5 iterations and we repeat this experiment for an increasing number of servers N. We empirically compute the average response time of undiscarded jobs, as a function of normalized arrival rate λ. We observe that the empirical curve approaches our analytical computation under the asymptotic independence conjecture, as the number of servers N increases. This provides an empirical validation of the asymptotic independence conjecture, and hence our theoretical results. In particular, it indicates that even for the most general of our policies, the asymptotic independence of queues is indeed true.

When the primary discard threshold is infinite, then all jobs get served. We illustrate a similar validation for two special cases where the primary replica is never discarded. The results for deterministic d replicas with no discard ($\pi(d, \infty, \infty)$ policy) is presented in Fig. B.11, and for deterministic d replicas with secondary on idle servers ($\pi(d, \infty, 0)$ policy) in Fig. B.12. The closed-form theoretical expressions of the conditional mean response time of these policies are provided in Remark 12 and Lemma 17 respectively. As in

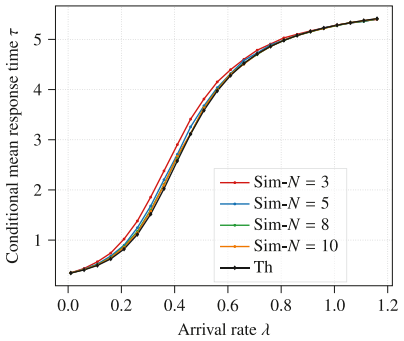


Fig. B.10. For the policy $\pi(d, T, T)$ with fixed thresholds $T_1 = T_2 = 5$, number of replicas $d = 3$,service rate $\mu = 1$, conditional mean response time τ as function of arrival rate λ for different values of servers $N \in \{3, 5, 8, 10\}$.

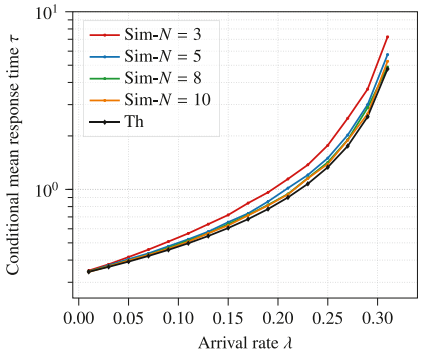


Fig. B.11. For the policy $\pi(d, \infty, \infty)$ with number of replicas $d = 3$, service rate $\mu = 1$, conditional mean response time τ as function of arrival rate λ for different values of servers $N \in \{3, 5, 8, 10\}$.

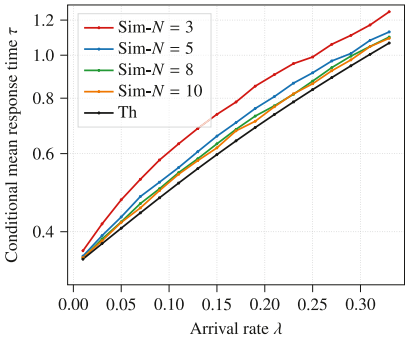


Fig. B.12. For the policy $\pi(d, \infty, 0)$ with number of replicas $d = 3$, service rate $\mu = 1$, conditional mean response time τ as function of arrival rate λ for different values of servers $N \in \{3, 5, 8, 10\}$.

the case of $\pi(d, T, T)$, we see that the empirically computed mean response time of undiscarded jobs converges to the corresponding theoretical expression with an increase in the number of servers N. This indicates that as the number of servers N increases the workload across queues tends to be independent, validating our conjecture on the asymptotic independence of queues. It is remarkable to note that the theoretical values and those obtained empirically from the simulation, coincide even when the number of servers N is as low as 10.

Even though, we have performed extensive validations for different values of T_1 and T_2 (for which closed-form results are available) and have observed similar behavior with an increase in the number of servers N, we have presented only a select few of the plots validating our models.

Appendix C. Proof of Theorem 7

Proof. From (3), we recall that the conditional mean response time for undiscarded jobs at stationarity is given by $\frac{\int_x \bar{H}(x)dx}{1-P_L}$, where $\bar{H}$ is the tail distribution for the response time of an undiscarded job in the system at stationarity, and defined in Definition 4. It suffices to show that for all $x \in \mathbb{R}_+$

$$\bar{H}(x) = (\bar{F}(T_1) + k(x,T_1))(\bar{F}(T_2) + k(x,T_2))^{d-1} - \bar{F}(T_1)\bar{F}(T_2)^{d-1}.$$

To this end, we recall that $I_{n,1}, I_{n,2}$ denote the disjoint random sets of servers where primary and secondary replicas for job n are dispatched. Further, the indicator that the nth job replica at server $j \in I_{n,1} \cup I_{n,2}$ with workload $W_{n,j}$ is not discarded is defined in (1). The set of servers where the nth job replicas are not discarded is denoted by $I_n = \{j \in I_{n,1} : \xi_{n,j} = 1\} \cup \{j \in I_{n,2} : \xi_{n,j} = 1\}$, and the indicator that the job n is not discarded is $\xi_n = \mathbb{1}_{\{I_n \neq \emptyset\}} = 1 - \prod_{j \in I_{n,1} \cup I_{n,2}} \xi_{n,j}$ as defined in (2). If the nth job is not discarded, then the indicator of the response time is larger than a threshold $x \in R_+$ is written as

$$\mathbb{1}_{\{R_n > x\}} = \xi_n \prod_{j \in I_n} \mathbb{1}_{\{W_{n,j} + X_{n,j} > x\}} = \xi_n \prod_{j \in I_{n,1}} (\xi_{n,j} \mathbb{1}_{\{W_{n,j} + X_{n,j} > x\}} + \bar{\xi}_{n,j}) \prod_{j \in I_{n,2}} (\xi_{n,j} \mathbb{1}_{\{W_{n,j} + X_{n,j} > x\}} + \bar{\xi}_{n,j}).$$

Substituting (2) for the indicator ξ_n in the above equation, using the fact that $\xi_{n,j}\bar{\xi}_{n,j} = 0$, and re-arranging the terms, we can write

$$\mathbb{1}_{\{R_n > x\}} = \prod_{j \in I_{n,1} \cup I_{n,2}} (\xi_j \mathbb{1}_{\{W_{n,j} + X_{n,j} > x\}} + \bar{\xi}_{n,j}) - \prod_{j \in I_{n,1} \cup I_{n,2}} \bar{\xi}_{n,j}.$$

Taking expectations on both sides of the above equations, using the independence of indicators $(\xi_{n,j} : j \in I_{n,1} \cup I_{n,2})$ with the limiting mean $\lim_{n\to\infty} \mathbb{E}[\xi_{n,j} \mathbb{1}_{\{j \in I_{n,1} \cup I_{n,2}\}} \Big| I_{n,1}, I_{n,2}] = \lim_{n\to\infty} [F(T_1)\gamma^1_{n,j} + F(T_2)\gamma^2_{n,j}]$, the definition of $k(x,T) = \lim_{n\to\infty} \mathbb{E}\left[\mathbb{1}_{\{W_{n,j} \leqslant T\}} \mathbb{1}_{\{X_{n,j} + W_{n,j} > x\}}\right]$, and the fact that all servers have identical limiting workload distribution F, we obtain the limiting tail distribution of the response time for an undiscarded job as

$$\lim_{n\to\infty} \mathbb{E}[\mathbb{1}_{\{R_n > x\}} \mid I_{n,1}, I_{n,2}] = (k(x,T_1) + \bar{F}(T_1))(k(x,T_2) + \bar{F}(T_2))^{d-1} - \bar{F}(T_1)\bar{F}(T_2)^{d-1}.$$

Since the right-hand side of the preceding equation does not depend on $I_{n,1}, I_{n,2}$, we have $\bar{H}(x) = \lim_{n\to\infty} \mathbb{E}[\mathbb{1}_{\{R_n > x\}} \mid I_{n,1}, I_{n,2}]$ as desired. □

Appendix D. Proof of Theorem 9

This section provides the moment-generating function-based approach for deriving the stationary workload distribution in a single queue in an N server system with *i.i.d.* service times and Poisson arrivals with threshold-based dispatching policy, $\pi(d, T_1, T_2)$. Although, the proof is provided only for the case where the service times are exponentially distributed with rate μ, the same approach can be used when the service times follow a shifted exponential distribution. We omit the details due to space constraints. Let us now begin the proof by providing two simple results.

Lemma 22. *For the interarrival time sequence $(Z_n : n \in \mathbb{N})$, we have*

$$\mathbb{E}\left[e^{\theta Z_{n+1}} \mathbb{1}_{\{W_n + X_n > Z_{n+1}\}} \mid W_n, X_n\right] = \frac{N\lambda}{N\lambda - \theta}(1 - e^{-(N\lambda - \theta)(W_n + X_n)}). \tag{D.1}$$

Proof. Recall that interarrival times $(Z_n : n \in \mathbb{N})$ are *i.i.d.* exponential with rate $N\lambda$, and duration Z_{n+1} is independent of past workloads $(W_1, \ldots, W_n)$ and past and present service times $(X_1, \ldots, X_n)$ for all $n \in \mathbb{Z}_+$. Hence, the result follows. □

Lemma 23. *For i.i.d. exponential service time sequence $(X_n : n \in \mathbb{N})$ with rate μ, we have*

$$\mathbb{E}\left[e^{-\theta X_n} \mathbb{1}_{\{X_n < T - W_n\}} \mid W_n\right] = \Phi_X(\theta)(1 - e^{-(\mu+\theta)(T - W_n)_+}). \tag{D.2}$$

In addition, we have the following identity

$$\frac{\Phi_X(\theta) - 1}{\theta} = -\frac{1}{\mu}\Phi_X(\theta). \tag{D.3}$$

Proof. The nth service time X_n is independent of workloads $(W_1, \ldots, W_n)$ seen by first n incoming arrivals. The first equality follows from this observation. The second equality follows from the fact that $\Phi_X(\theta) = \frac{\mu}{\mu+\theta}$. □

Proposition 24. *For an N server system with i.i.d. exponential service times of rate μ, Poisson arrivals of rate $N\lambda$ under $\pi(d, T_1, T_2)$ policy and the moment generating functions of the limiting workload W in a single queue defined in Definition 8,*

$$\begin{aligned}\Phi_W(\theta) = {} & F(0)(1 + \frac{\bar{\lambda}}{\theta + \mu - \bar{\lambda}}) + \big((\mu - \lambda)\bar{F}(T_2) + \lambda\bar{F}(T_1)\big)e^{-\theta T_2}\left[\frac{1}{\theta + \mu - \lambda} - \frac{1}{\theta + \mu - \bar{\lambda}}\right] \\ & - \bar{F}(T_1)\mu e^{-\theta T_1}\left[\frac{1}{\theta + \mu - \lambda} - \frac{1}{\theta + \mu}\right].\end{aligned} \tag{D.4}$$

Proof. For each $n \in \mathbb{N}$, we denote the restricted moment generating function for W_n as

$$\Phi_{W_n}(\theta) = \mathbb{E}\left[e^{-\theta W_n}\right], \quad \Phi_{1,n}(\theta) = \mathbb{E}\left[e^{-\theta W_n}\mathbb{1}_{\{W_n>T_1\}}\right], \quad \Phi_{2,n}(\theta) = \mathbb{E}\left[e^{-\theta W_n}\mathbb{1}_{\{W_n>T_2\}}\right]. \tag{D.5}$$

The moment generating function for the workload at $(n+1)$th arrival is given by

$$\Phi_{W_{n+1}}(\theta) = \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_1\}}\right] + \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{T_2<W_n\leqslant T_1\}}\right] + \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n\leqslant T_2\}}\right].$$

We can explicitly re-write (5), in the following three regions

$$W_{n+1} = \begin{cases} (W_n - Z_{n+1})_+, & W_n \in (T_1, \infty), \\ (1-\gamma_n^1)(W_n - Z_{n+1})_+ + \gamma_n^1(W_n + X_n - Z_{n+1})_+, & W_n \in (T_2, T_1], \\ (1-\gamma_n^1-\gamma_n^2)(W_n - Z_{n+1})_+ + (\gamma_n^1+\gamma_n^2)(W_n + X_n - Z_{n+1})_+, & W_n \in [0, T_2]. \end{cases} \tag{D.6}$$

This relation allows us to find the moment generating function for W_{n+1} in terms of restricted moment generating functions for W_n.

Step 1: We first observe that in the region $W_n > T_1$, we have $W_{n+1} = (W_n - Z_{n+1})\mathbb{1}_{\{W_n>Z_{n+1}\}}$ from (D.6). From the tower property of conditional expectation, we can write

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_1\}}\right] = \mathbb{E}\left[\mathbb{1}_{\{W_n>T_1\}}\mathbb{E}\left[\mathbb{1}_{\{W_n\leqslant Z_{n+1}\}} \mid W_n\right]\right] + \mathbb{E}\left[e^{-\theta W_n}\mathbb{1}_{\{W_n>T_1\}}\mathbb{E}\left[e^{\theta Z_{n+1}}\mathbb{1}_{\{W_n>Z_{n+1}\}} \mid W_n\right]\right].$$

Since Z_{n+1} is exponential with rate $N\lambda$ and independent of W_n, we get $\mathbb{E}\left[\mathbb{1}_{\{W_n\leqslant Z_{n+1}\}} \mid W_n\right] = e^{-\mathbb{N}\lambda W_n}$, and from the identity in (D.1) for $X_n = 0$, we get $\mathbb{E}\left[e^{\theta Z_{n+1}}\mathbb{1}_{\{W_n>Z_{n+1}\}} \mid W_n\right] = \frac{N\lambda}{N\lambda-\theta}(1 - e^{-(N\lambda-\theta)(W_n)})$. Substituting these in the above equation and from the definition of Φ_1, we obtain

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_1\}}\right] = \mathbb{E}\left[\mathbb{1}_{\{W_n>T_1\}}e^{-N\lambda W_n}\right] + \frac{N\lambda}{N\lambda-\theta}\mathbb{E}\left[\mathbb{1}_{\{W_n>T_1\}}(e^{-\theta W_n} - e^{-N\lambda W_n})\right] = \frac{N\lambda\Phi_{1,n}(\theta) - \theta\Phi_{1,n}(N\lambda)}{N\lambda-\theta}. \tag{D.7}$$

Step 2: We next observe that in the region $W_n \in (T_2, T_1]$, we have $W_{n+1} = (W_n - Z_{n+1})\mathbb{1}_{\{W_n>Z_{n+1}\}}$ with probability $1 - \frac{1}{N}$, and $W_{n+1} = (W_n + X_n - Z_{n+1})\mathbb{1}_{\{W_n+X_n>Z_{n+1}\}}$ with probability $\frac{1}{N}$. We can write

$$\begin{aligned}\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{(T_2,T_1]}(W_n)\right] &= \left(1-\frac{1}{N}\right)\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in(T_2,T_1]\}}\mathbb{E}\left[\mathbb{1}_{\{W_n\leqslant Z_{n+1}\}} \mid W_n\right]\right] + \mathbb{E}\left[e^{-\theta W_n}\mathbb{1}_{\{W_n\in(T_2,T_1]\}}\mathbb{E}\left[e^{\theta Z_{n+1}}\mathbb{1}_{\{W_n>Z_{n+1}\}} \mid W_n\right]\right]\right) \\ &+ \frac{1}{N}\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in(T_2,T_1]\}}\mathbb{E}\left[\mathbb{1}_{\{W_n+X_n\leqslant Z_{n+1}\}} \mid W_n, X_n\right]\right] + \mathbb{E}\left[e^{-\theta(W_n+X_n)}\mathbb{1}_{\{W_n\in(T_2,T_1]\}}\mathbb{E}\left[e^{\theta Z_{n+1}}\mathbb{1}_{\{W_n+X_n>Z_{n+1}\}} \mid W_n, X_n\right]\right]\right).\end{aligned}$$

Using the fact that Z_{n+1} is exponential with rate $N\lambda$ and independent of W_n, the identity in (D.1), we get

$$\begin{aligned}\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{(T_2,T_1]}(W_n)\right] &= \frac{(N-1)}{N}\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in(T_2,T_1]\}}e^{-N\lambda W_n}\right] + \frac{N\lambda}{N\lambda-\theta}\mathbb{E}\left[\mathbb{1}_{\{W_n\in(T_2,T_1]\}}(e^{-\theta W_n} - e^{-N\lambda W_n})\right]\right) \\ &+ \frac{1}{N}\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in(T_2,T_1]\}}e^{-N\lambda(W_n+X_n)}\right] + \frac{N\lambda}{N\lambda-\theta}\mathbb{E}\left[\mathbb{1}_{\{W_n\in(T_2,T_1]\}}(e^{-\theta(W_n+X_n)} - e^{-N\lambda(W_n+X_n)})\right]\right).\end{aligned}$$

Substituting the definition of restricted moment generating functions for W_n and moment generating functions for X_n in the above equation, we get

$$\begin{aligned}\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{(T_2,T_1]}(W_n)\right] &= \frac{(N-1)}{N(N\lambda-\theta)}\Big(-\theta(\Phi_{2,n}(N\lambda) - \Phi_{1,n}(N\lambda)) + N\lambda(\Phi_{2,n}(\theta) - \Phi_{1,n}(\theta))\Big) \\ &+ \frac{1}{N(N\lambda-\theta)}\Big(-\theta\Phi_X(N\lambda)(\Phi_{2,n}(N\lambda) - \Phi_{1,n}(N\lambda)) + N\lambda\Phi_X(\theta)(\Phi_{2,n}(\theta) - \Phi_{1,n}(\theta))\Big).\end{aligned} \tag{D.8}$$

Since $\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_2\}}\right] = \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_1\}}\right] + \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{T_1\geqslant W_n>T_2\}}\right]$, summing (D.7) and (D.8), we obtain

$$\begin{aligned}\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_2\}}\right] &= \left[-\frac{\theta}{N\lambda-\theta}\Phi_{2,n}(N\lambda) + \frac{N\lambda}{N\lambda-\theta}\Phi_{2,n}(\theta)\right] \\ &+ \frac{\lambda\theta}{N\lambda-\theta}\left[-(\Phi_{2,n}(N\lambda) - \Phi_{1,n}(N\lambda))\frac{(\Phi_X(N\lambda)-1)}{N\lambda} + (\Phi_{2,n}(\theta) - \Phi_{1,n}(\theta))\frac{(\Phi_X(\theta)-1)}{\theta}\right].\end{aligned} \tag{D.9}$$

Step 3: We next observe that in the region $W_n \leqslant T_2$, we have $W_{n+1} = (W_n - Z_{n+1})\mathbb{1}_{\{W_n>Z_{n+1}\}}$ with probability $1 - \frac{\bar{\lambda}}{N\lambda}$, and $W_{n+1} = (W_n + X_n - Z_{n+1})\mathbb{1}_{\{W_n+X_n>Z_{n+1}\}}$ with probability $\frac{\bar{\lambda}}{N\lambda}$. We can write

$$\begin{aligned}\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{[0,T_2]}(W_n)\right] &= \left(1-\frac{\bar{\lambda}}{N\lambda}\right)\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in[0,T_2]\}}\mathbb{E}\left[\mathbb{1}_{\{W_n\leqslant Z_{n+1}\}} \mid W_n\right]\right] + \mathbb{E}\left[e^{-\theta W_n}\mathbb{1}_{\{W_n\in[0,T_2]\}}\mathbb{E}\left[e^{\theta Z_{n+1}}\mathbb{1}_{\{W_n>Z_{n+1}\}} \mid W_n\right]\right]\right) \\ &+ \frac{\bar{\lambda}}{N\lambda}\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in[0,T_2]\}}\mathbb{E}\left[\mathbb{1}_{\{W_n+X_n\leqslant Z_{n+1}\}} \mid W_n, X_n\right]\right] + \mathbb{E}\left[e^{-\theta(W_n+X_n)}\mathbb{1}_{\{W_n\in[0,T_2]\}}\mathbb{E}\left[e^{\theta Z_{n+1}}\mathbb{1}_{\{W_n+X_n>Z_{n+1}\}} \mid W_n, X_n\right]\right]\right).\end{aligned}$$

Using the fact that Z_{n+1} is exponential with rate $N\lambda$ and independent of W_n, the identity in (D.1), we get

$$\begin{aligned}\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{[0,T_2]}(W_n)\right] &= \frac{(N\lambda-\bar{\lambda})}{N\lambda}\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in[0,T_2]\}}e^{-N\lambda W_n}\right] + \frac{N\lambda}{N\lambda-\theta}\mathbb{E}\left[\mathbb{1}_{\{W_n\in[0,T_2]\}}(e^{-\theta W_n} - e^{-N\lambda W_n})\right]\right) \\ &+ \frac{\bar{\lambda}}{N\lambda}\left(\mathbb{E}\left[\mathbb{1}_{\{W_n\in[0,T_2]\}}e^{-N\lambda(W_n+X_n)}\right] + \frac{N\lambda}{N\lambda-\theta}\mathbb{E}\left[\mathbb{1}_{\{W_n\in[0,T_2]\}}\left(e^{-\theta(W_n+X_n)} - e^{-N\lambda(W_n+X_n)}\right)\right]\right).\end{aligned}$$

Substituting the definition of restricted moment generating functions for W_n and moment generating function for X_n in the above equation, we get

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{[0,T_2]}(W_n)\right] = \frac{(N\lambda-\bar{\lambda})}{N\lambda(N\lambda-\theta)}\Big(-\theta(\Phi_{W_n}(N\lambda)-\Phi_{2,n}(N\lambda)) + N\lambda(\Phi_{W_n}(\theta)-\Phi_{2,n}(\theta))\Big) + \frac{\bar{\lambda}}{N\lambda(N\lambda-\theta)}\Big(-\theta\Phi_X(N\lambda)(\Phi_{W_n}(N\lambda)-\Phi_{2,n}(N\lambda)) + N\lambda\Phi_X(\theta)(\Phi_{W_n}(\theta)-\Phi_{2,n}(\theta))\Big). \tag{D.10}$$

Since $\Phi_{W_{n+1}}(\theta) = \mathbb{E}\left[e^{-\theta W_{n+1}}\right] = \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n>T_2\}}\right] + \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_n\leqslant T_2\}}\right]$, summing (D.9) and (D.10), we obtain

$$\begin{aligned}\Phi_{W_{n+1}}(\theta) = &\Big[-\frac{\theta}{N\lambda-\theta}\Phi_{W_n}(N\lambda) + \frac{N\lambda}{N\lambda-\theta}\Phi_{W_n}(\theta)\Big] \\ &+ \frac{\lambda\theta}{N\lambda-\theta}\Big[-(\Phi_{2,n}(N\lambda)-\Phi_{1,n}(N\lambda))\frac{(\Phi_X(N\lambda)-1)}{N\lambda} + (\Phi_{2,n}(\theta)-\Phi_{1,n}(\theta))\frac{(\Phi_X(\theta)-1)}{\theta}\Big] \\ &+ \frac{\bar{\lambda}\theta}{N\lambda-\theta}\Big[-(\Phi_{W_n}(N\lambda)-\Phi_{2,n}(N\lambda))\frac{(\Phi_X(N\lambda)-1)}{N\lambda} + (\Phi_{W_n}(\theta)-\Phi_{2,n}(\theta))\frac{(\Phi_X(\theta)-1)}{\theta}\Big].\end{aligned} \tag{D.11}$$

Step 4: As $n\to\infty$, the limiting distribution of W_n is given by F, and therefore, we have

$$\Phi_W(\theta) = \lim_{n\to\infty}\Phi_{W_n}(\theta), \quad \Phi_2(\theta) = \lim_{n\to\infty}\Phi_{2,n}(\theta), \quad \Phi_1(\theta) = \lim_{n\to\infty}\Phi_{1,n}(\theta).$$

Rearranging terms in (D.11), we get

$$\begin{aligned}&\Phi_W(N\lambda) + \Big[\lambda(\Phi_2(N\lambda)-\Phi_1(N\lambda)) + \bar{\lambda}(\Phi_W(N\lambda)-\Phi_2(N\lambda))\Big]\frac{(\Phi_X(N\lambda)-1)}{N\lambda} \\ &= \Phi_W(\theta) + \Big[\lambda(\Phi_2(\theta)-\Phi_1(\theta)) + \bar{\lambda}(\Phi_W(\theta)-\Phi_2(\theta))\Big]\frac{(\Phi_X(\theta)-1)}{\theta}.\end{aligned}$$

We observe that LHS and RHS have the form $f(\theta) = f(N\lambda)$ for an arbitrary function f and variables θ and λ. Therefore, we conclude that $f(\theta) = f(0)$. Further, note that $\Phi_i(0) = \bar{F}_{T_i}$ for $i\in[2]$. Then, using Eq. (D.3), we can write for exponential service times,

$$\Phi_W(\theta)\Big(1-\frac{\bar{\lambda}}{\mu}\Phi_X(\theta)\Big) + \Big[\frac{\bar{\lambda}-\lambda}{\mu}\Phi_2(\theta) + \frac{\lambda}{\mu}\Phi_1(\theta)\Big]\Phi_X(\theta) = 1-\frac{\bar{\lambda}}{\mu} + \Big[\frac{\bar{\lambda}-\lambda}{\mu}\bar{F}(T_2) + \frac{\lambda}{\mu}\bar{F}(T_1)\Big].$$

Now, we substitute $\Phi_1(\theta)$ and $\Phi_2(\theta)$ from Eqs. (D.12) and (D.15) respectively in the above equation. Further incorporating Eqs. (D.13) and (D.16) and rearranging the terms will yield Eq. (D.4). □

Remark 16. Upon inverting the moment generating function in Eq. (D.4), we see that the complementary workload distribution function for $w\geqslant 0$ is given by

$$\begin{aligned}\bar{F}(w) = &1 - F(0)\Big(1+\frac{\bar{\lambda}(1-e^{-(\mu-\bar{\lambda})w})}{\mu-\bar{\lambda}}\Big) + \mu\bar{F}(T_1)\Big(\frac{(1-e^{-(\mu-\lambda)(w-T_1)_+})}{\mu-\lambda} - \frac{(1-e^{-\mu(w-T_1)_+})}{\mu}\Big) \\ &- ((\mu-\lambda)\bar{F}(T_2) + \lambda\bar{F}(T_1))\Big(\frac{(1-e^{-(\mu-\lambda)(w-T_2)_+})}{\mu-\lambda} - \frac{(1-e^{-(\mu-\bar{\lambda})(w-T_2)_+})}{\mu-\bar{\lambda}}\Big).\end{aligned}$$

In addition, we can find the constant,

$$F(0) = 1-\frac{\bar{\lambda}}{\mu} + \Big[\frac{\bar{\lambda}-\lambda}{\mu}\bar{F}(T_2) + \frac{\lambda}{\mu}\bar{F}(T_1)\Big].$$

Proposition 25. *For an N server system with i.i.d. exponential service times of rate μ, Poisson arrivals of rate $N\lambda$ under $\pi(d,T_1,T_2)$ policy and the moment generating functions of the limiting workload W in a single queue defined in Definition 8,*

$$\Phi_1(\theta) = e^{-\mu T_1}\Big[\frac{\lambda}{\mu}(\Phi_2(-\mu)-\Phi_1(-\mu)) + \frac{\bar{\lambda}}{\mu}(\Phi(-\mu)-\Phi_2(-\mu))\Big]e^{-\theta T_1}\Phi_X(\theta). \tag{D.12}$$

This implies that for $w>T_1$, $\bar{F}(w) = \bar{F}(T_1)e^{-\mu(w-T_1)_+}$, where

$$\bar{F}(T_1) = e^{-\mu T_1}\Big[\frac{\lambda}{\mu}(\Phi_2(-\mu)-\Phi_1(-\mu)) + \frac{\bar{\lambda}}{\mu}(\Phi(-\mu)-\Phi_2(-\mu))\Big]. \tag{D.13}$$

Proof. The computation remains similar to the previous step, with an additional restriction of $W_{n+1}>T_1$. Therefore, we can write

$$\Phi_{1,n+1}(\theta) = \mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_1\}}\Big(\mathbb{1}_{\{W_n>T_1\}} + \mathbb{1}_{\{T_2<W_n\leqslant T_1\}} + \mathbb{1}_{\{W_n\leqslant T_2\}}\Big)\right]. \tag{D.14}$$

We sequentially compute the first term, the summation of the first two terms, and the summation of all three terms as before. In the region $W_n>T_1$, we have $e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_1\}}\mathbb{1}_{\{W_n>T_1\}} = e^{-\theta(W_n-Z_{n+1})}\mathbb{1}_{\{Z_{n+1}<W_n-T_1\}}\mathbb{1}_{\{W_n>T_1\}}$. Then, it follows that

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_1\}}\mathbb{1}_{\{W_n>T_1\}}\right] = \frac{N\lambda}{N\lambda-\theta}\mathbb{E}\left[e^{-\theta W_n}\mathbb{1}_{\{W_n>T_1\}}(1-e^{-(N\lambda-\theta)(W_n-T_1)})\right] = \frac{N\lambda}{N\lambda-\theta}\Big(\Phi_{1,n}(\theta) - e^{(N\lambda-\theta)T_1}\Phi_{1,n}(N\lambda)\Big).$$

Note that, in the region $W_n \leqslant T_1$, it is not possible for $W_{n+1} > T_1$, unless the nth arrival with service time X_n is admitted at the cavity queue. This occurs with probability $\frac{1}{N}$ in region $T_2 < W_n \leqslant T_1$, and with probability $\frac{\bar{\lambda}}{N\lambda}$ in region $W_n \leqslant T_2$. Therefore, for the region $T_2 < W_n \leqslant T_1$, we can write

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_1\}}\mathbb{1}_{\{T_2<W_n\leqslant T_1\}}\right] = \frac{\lambda e^{-(\mu+\theta)T_1}}{N\lambda-\theta}(\Phi_{2,n}(-\mu)-\Phi_{1,n}(-\mu))(\Phi_X(\theta)-\Phi_X(N\lambda)).$$

Similarly, for the region $W_n \leqslant T_2$, we can write

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_1\}}\mathbb{1}_{\{W_n\leqslant T_2\}}\right] = \frac{\bar{\lambda} e^{-(\mu+\theta)T_1}}{N\lambda-\theta}(\Phi_{W_n}(-\mu)-\Phi_{2,n}(-\mu))(\Phi_X(\theta)-\Phi_X(N\lambda)).$$

Substituting the above three equations in Eq. (D.14), taking the limit of moment generating functions, and rearranging the terms as in the previous proof, we get

$$\Phi_1(\theta) = \left[\lambda(\Phi_2(-\mu)-\Phi_1(-\mu)) + \bar{\lambda}(\Phi_W(-\mu)-\Phi_2(-\mu))\right]\frac{e^{-(\mu+\theta)T_1}}{\mu}\Phi_X(\theta).$$

The result follows by inverting the moment generating function and noting that $\Phi_1(0) = \bar{F}(T_1)$. □

Proposition 26. *For an N server system with i.i.d. exponential service times of rate μ, Poisson arrivals of rate $N\lambda$ under $\pi(d, T_1, T_2)$ policy and the moment generating functions of the limiting workload W in a single queue defined in Definition 8,*

$$\Phi_2(\theta) = \frac{\lambda}{\mu}(\Phi_2(\theta)-\Phi_1(\theta))\Phi_X(\theta) + \frac{\bar{\lambda}}{\mu}e^{-\mu T_2}\left(\Phi(-\mu)-\Phi_2(-\mu)\right)e^{-\theta T_2}\Phi_X(\theta). \tag{D.15}$$

This implies that for $w > T_2$, $\bar{F}(w) = \bar{F}(T_1)\left(e^{-\mu(w-T_1)_+} - \frac{\mu}{\mu-\lambda}e^{-(\mu-\lambda)(w-T_1)_+}\right) + \left[e^{-\mu T_2}(\Phi(-\mu)-\Phi_2(-\mu))\right]\frac{\bar{\lambda}}{\mu-\lambda}e^{-(\mu-\lambda)(w-T_2)_+}$. In addition,

$$\bar{F}(T_2) = \frac{\bar{\lambda}}{\mu-\lambda}e^{-\mu T_2}(\Phi(-\mu)-\Phi_2(-\mu)) - \frac{\lambda}{\mu-\lambda}\bar{F}(T_1). \tag{D.16}$$

Proof. The computation remains similar to the previous case but here we have the restriction of $W_{n+1} > T_2$. Then, we can write

$$\Phi_{2,n+1}(\theta) = \mathbb{E}[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_2\}}\left(\mathbb{1}_{\{W_n>T_2\}} + \mathbb{1}_{\{T_2<W_n\leqslant T_1\}} + \mathbb{1}_{\{W_n\leqslant T_2\}}\right)].$$

We sequentially compute the first term, the summation of the first two terms, and the summation of all three terms as before. The indicator $W_{n+1} > T_2$ implies that W_{n+1} cannot be zero. In the region $W_n > T_1$, we have $e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_2\}}\mathbb{1}_{\{W_n>T_1\}} = e^{-\theta(W_n-Z_{n+1})}\mathbb{1}_{\{Z_{n+1}<W_n-T_2\}}\mathbb{1}_{\{W_n>T_1\}}$. Therefore, it follows that

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_2\}}\mathbb{1}_{\{W_n>T_1\}}\right] = \frac{N\lambda\left(\Phi_{1,n}(\theta) - e^{(N\lambda-\theta)T_2}\Phi_{1,n}(N\lambda)\right)}{N\lambda-\theta}.$$

Similarly, for the region $T_2 < W_n \leqslant T_1$, an external arrival is admitted with probability $\frac{1}{N}$. When there is no arrival $W_{n+1} = W_n - Z_{n+1}$, and we have

$$\mathbb{E}\left[e^{-\theta(W_n-Z_{n+1})}\mathbb{1}_{\{W_n-X_{n+1}>T_2\}}\mathbb{1}_{\{T_2<W_n\leqslant T_1\}}\right] = \frac{N\lambda\left(\Phi_{2,n}(\theta)-\Phi_{1,n}(\theta) - e^{(N\lambda-\theta)T_2}(\Phi_{2,n}(N\lambda)-\Phi_{1,n}(N\lambda))\right)}{N\lambda-\theta}.$$

In the region $T_2 < W_n \leqslant T_1$, the nth arrival with service time X_n is admitted at the cavity queue with probability $\frac{1}{N}$. In this case, $W_{n+1} = W_n + X_n - Z_{n+1}$, and we can write

$$\mathbb{E}\left[e^{-\theta(W_n+X_n-Z_{n+1})}\mathbb{1}_{\{W_n+X_n-Z_{n+1}>T_2\}}\mathbb{1}_{\{T_2<W_n\leqslant T_1\}}\right] = \frac{N\lambda}{N\lambda-\theta}\left[(\Phi_{2,n}(\theta)-\Phi_{1,n}(\theta))\Phi_X(\theta) - e^{(N\lambda-\theta)T_2}(\Phi_{2,n}(N\lambda)-\Phi_{1,n}(N\lambda))\Phi_X(N\lambda)\right].$$

Combining these results in the region $W_n > T_2$, we can write

$$\begin{aligned}\mathbb{E}\left[e^{-\theta(W_{n+1})}\mathbb{1}_{\{W_{n+1}>T_2\}}\mathbb{1}_{\{W_n>T_2\}}\right] =& \frac{N\lambda\left[\Phi_{2,n}(\theta) - e^{(N\lambda-\theta)T_2}\Phi_{2,n}(N\lambda)\right]}{N\lambda-\theta} \\ &+ \frac{\lambda}{N\lambda-\theta}\left[(\Phi_{2,n}(\theta)-\Phi_{1,n}(\theta))(\Phi_X(\theta)-1) - e^{(N\lambda-\theta)T_2}(\Phi_{2,n}(N\lambda)-\Phi_{1,n}(N\lambda))(\Phi_X(N\lambda)-1)\right].\end{aligned}$$

In the region $W_n \leqslant T_2$, it is not possible for $W_{n+1} > T_2$, unless the n arrival with service time X_n is admitted at the cavity queue. This occurs with probability $\frac{\bar{\lambda}}{N\lambda}$, and we can write

$$\mathbb{E}\left[e^{-\theta W_{n+1}}\mathbb{1}_{\{W_{n+1}>T_1\}}\mathbb{1}_{\{W_n\leqslant T_2\}}\right] = \frac{\bar{\lambda}e^{-(\mu+\theta)T_2}}{N\lambda-\theta}(\Phi_{W_n}(-\mu)-\Phi_{2,n}(-\mu))(\Phi_X(\theta)-\Phi_X(N\lambda)).$$

Combining the above equations, taking the limit of moment generating functions, and rearranging the terms as in previous proofs, we obtain

$$\Phi_2(\theta) = \frac{\lambda}{\mu}(\Phi_2(\theta)-\Phi_1(\theta))\Phi_X(\theta) + \frac{\bar{\lambda}}{\mu}e^{-\mu T_2}(\Phi(-\mu)-\Phi_2(-\mu))e^{-\theta T_2}\Phi_X(\theta). \tag{D.17}$$

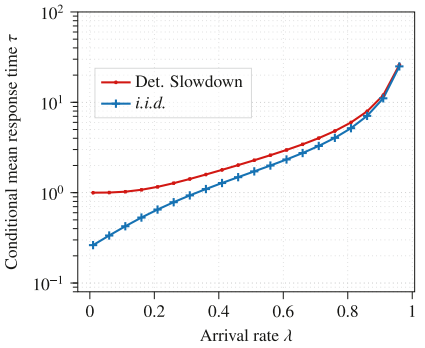


Fig. E.13. For the deterministic slowdown with slowdown factor $s = 1$ and *i.i.d.* service times under $\pi(d, \infty, 0)$ policy with number of replicas $d = 4$, service rate $\mu = 1$ and number of servers $N = 20$, conditional mean response time τ as function of arrival rate λ.

To prove the second statement, note that $\Phi_1(\theta) = \bar{F}(T_1)e^{-\theta T_1}\Phi_X(\theta)$ from Eq. (D.12). Substitution and simplification tell us that $\Phi_2(\theta) = \left(\frac{1}{\mu+\theta} - \frac{1}{\mu-\lambda+\theta}\right)\mu\bar{F}(T_1)e^{-\theta T_1} + \frac{\bar{\lambda}e^{-\theta T_2}}{(\mu-\lambda+\theta)}e^{-\mu T_2}(\Phi(-\mu) - \Phi_2(-\mu))$ when service times are exponentially distributed with rate μ. The result follows by inverting this moment generating function and the fact that $\Phi_2(0) = \bar{F}(T_2)$. □

Appendix E. Mean response time under identical replicas

In this appendix, we analyze the performance of our proposed policy when the service time distribution follows a special case of the S&X model. We assume that the slowdown factor takes a deterministic value $S_i = s$ for all servers $i \in [N]$ and some finite $s \geqslant 1$ and the job service time X_n is *i.i.d.* exponential with rate μ. Therefore, the service time of the nth job at all the servers at which it gets accepted for processing will be identical and is a realization of the scaled exponential random variable sX_n. For this service model, we will derive the mean response time of a job for the $\pi(d, T_1, T_2)$ policy under the assumption of asymptotic independence among the workloads at different queues.

Remark 17. We observe that the mean workload at the cavity queue under this model remains identical to the case when the job sizes are *i.i.d.* exponential. Therefore, the loss probability for this model will remain identical to the case when the job sizes are *i.i.d.* exponential and is given by Lemma 6.

Lemma 27. *The conditional mean response time of a job under the $\pi(d, T_1, T_2)$ policy and identical job replica size of mean s/μ is given by $\tau = \frac{1}{1-P_L}\left[\int_0^{T_2}\bar{F}(w)^d - \bar{F}(T_1)\bar{F}(T_2)^{d-1}dw + \int_{T_2}^{T_1}(\bar{F}(w) - \bar{F}(T_1))\bar{F}(T_2)^{d-1}dw\right] + \frac{s}{\mu}$ where P_L is the loss probability and $\bar{F}(w)$ is the marginal complementary workload distribution at equilibrium.*

Proof. Consider a job arriving at the set of d randomly selected set of primary and secondary servers, I_1 and I_2. Note that the job gets admitted only at a set of servers $I \subseteq I_1 \cup I_2$. Suppose that the current workload at server j is denoted by W_j and the indicator of a job being undiscarded by $\xi = \mathbb{1}_{\{I \neq \emptyset\}}$ as defined previously. Then, the response time of an undiscarded job is given by $R = Z + \xi sX$, where we define $Z \triangleq \xi\left(\min\left\{W_j : j \in I\right\}\right)$ and the mean response time is

$$\mathbb{E}[R] = \mathbb{E}[Z] + \frac{s(1-P_L)}{\mu}. \tag{E.1}$$

Next, we observe that $\mathbb{1}_{\{Z>z\}} = \xi\mathbb{1}_{\{\min\{W_j : j\in I\}>z\}} = \xi\left(\prod_{I_1\cup I_2}(\xi_j\mathbb{1}_{\{W_j>z\}} + \bar{\xi}_j)\right)$. From the independence of the workloads and therefore of the indicators ξ_j across queues and by substituting for ξ from Eq. (2), we get

$$\begin{aligned}\mathbb{E}[\mathbb{1}_{\{Z>z\}}] &= \mathbb{E}[\prod_{I_1\cup I_2}(\xi_j\mathbb{1}_{\{W_j>z\}} + \bar{\xi}_j)] - \mathbb{E}[\prod_{I_1\cup I_2}\bar{\xi}_j] = \prod_{I_1\cup I_2}\mathbb{E}[(\xi_j\mathbb{1}_{\{W_j>z\}} + \bar{\xi}_j)] - \prod_{I_1\cup I_2}\mathbb{E}[\bar{\xi}_j] \\ &= (\bar{F}(z)^d - \bar{F}(T_1)\bar{F}(T_2)^{d-1})\mathbb{1}_{\{z\leqslant T_2\}} + (\bar{F}(z) - \bar{F}(T_1))\bar{F}(T_2)^{d-1}\mathbb{1}_{\{T_2<z\leqslant T_1\}}.\end{aligned}$$

Since $\mathbb{E}[Z] = \int_0^\infty P\{Z > z\}\,dz$, we obtain $\mathbb{E}[Z] = \int_0^{T_2}\bar{F}(z)^d - \bar{F}(T_1)\bar{F}(T_2)^{d-1}dz + \int_{T_2}^{T_1}(\bar{F}(z) - \bar{F}(T_1))\bar{F}(T_2)^{d-1}dz$. The result follows from Eqs. (E.1) and (3). □

Corollary 28. *For the special case of replication on idle secondary servers in N server system with Poisson arrivals of rate λN, the conditional mean response time under deterministic slowdown and exponential job sizes with rate μ is given by $\tau = \bar{F}(0)^d + \int_{0^+}^\infty \bar{F}(z)\bar{F}(0)^{d-1}dz + \frac{s}{\mu}$ where $F(0) = \frac{(1-\frac{\lambda}{\mu})(1-\frac{\bar{\lambda}}{\mu})}{(1-\frac{\lambda}{\mu})+\frac{\bar{\lambda}}{\mu}(\frac{\lambda}{\mu}-\frac{\bar{\lambda}}{\mu})}$ and $\bar{F}(z) = 1 - F(0)\left[1 - \frac{\bar{\lambda}}{\mu-\lambda}(1-e^{-(\mu-\lambda)z})\right]$, $z > 0$.*

We provide a comparison of the mean response time for the $\pi(d, \infty, 0)$ policy under deterministic slowdown and *i.i.d.* exponential service times in Fig. E.13. We observe that for low arrival rates, the performance is comparatively worse when the service times are identical but not independent. However, the performance of both models converges for higher arrival rates as the chances of secondary replicas getting admitted at the servers diminish with an increase in arrival rate.

References

[1] W. Winston, Optimality of the shortest line discipline, J. Appl. Probab. 14 (1) (1977) 181–189.
[2] M. Mitzenmacher, The power of two choices in randomized load balancing, IEEE Trans. Parallel Distrib. Syst. 12 (10) (2001) 1094–1104.
[3] N.D. Vvedenskaya, R.L. Dobrushin, F.I. Karpelevich, Queueing system with selection of the shortest of two queues: An asymptotic approach, Problemy Peredachi Informatsii 32 (1) (1996) 20–34.
[4] T. Hellemans, B.V. Houdt, On the power-of-d-choices with least loaded server selection, Proc. ACM Meas. Anal. Comput. Syst. (PACM) 2 (2) (2018) 1–22.
[5] U. Ayesta, On redundancy-d with cancel-on-start a.k.a join-shortest-work (d), ACM SIGMETRICS Perform. Eval. Rev. 46 (2) (2019) 24–26.
[6] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M.H. Yahia, M. Zhang, Congestion control for large-scale RDMA deployments, SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 523–536.
[7] M. van der Boor, S. Borst, J. van Leeuwaarden, Hyper-scalable JSQ with sparse feedback, Proc. ACM Meas. Anal. Comput. Syst. (PACM) 3 (1) (2019) 1–37.
[8] M. van der Boor, M. Zubeldia, S. Borst, Zero-wait load balancing with sparse messaging, Oper. Res. Lett. 48 (3) (2020) 368–375.
[9] R. Jinan, P. Parag, H. Tyagi, Tracking an auto-regressive process with limited communication, in: IEEE Inter. Symp. Info. Theory (ISIT), 2020, pp. 2462–2467.
[10] R. Jinan, P. Parag, H. Tyagi, Tracking an auto-regressive process with limited communication per unit time, Entropy 23 (3) (2021) 347.
[11] G. Mendelson, K. Xu, CARE: Resource allocation using sparse communication, 2022, arXiv:2206.02410.
[12] S. Vargaftik, I. Keslassy, A. Orda, LSQ: Load balancing in large-scale heterogeneous systems with multiple dispatchers, IEEE/ACM Trans. Netw. 28 (3) (2020) 1186–1198.
[13] X. Zhou, N. Shroff, A. Wierman, Asymptotically optimal load balancing in large-scale heterogeneous systems with multiple dispatchers, Perform. Eval. 145 (2021) 102146.
[14] Y. Lu, Q. Xie, G. Kliot, A. Geller, J.R. Larus, A. Greenberg, Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services, Perform. Eval. 68 (11) (2011) 1056–1071.
[15] D. Gamarnik, J.N. Tsitsiklis, M. Zubeldia, Delay, memory, and messaging tradeoffs in distributed service systems, Stoch. Syst. 8 (1) (2018) 45–74.
[16] T. Hellemans, B.V. Houdt, Performance analysis of load balancing policies with memory, Perform. Eval. 153 (2022) 102259.
[17] U. Ayesta, T. Bodas, I.M. Verloop, On a unifying product form framework for redundancy models, Perform. Eval. 127–128 (2018) 93–119.
[18] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, M. Velednitsky, S. Zbarsky, Redundancy-d: The power of d choices for redundancy, Oper. Res. 65 (4) (2017) 1078–1094.
[19] K. Gardner, R. Righter, Product forms for FCFS queueing models with arbitrary server-job compatibilities: an overview, Queueing Syst. 96 (1–2) (2020) 3–51.
[20] U. Ayesta, T. Bodas, J.-P.L. Dorsman, I.M. Verloop, A token-based central queue with order-independent service rates, Oper. Res. 70 (1) (2022) 545–561.
[21] K. Lee, R. Pedarsani, K. Ramchandran, On scheduling redundant requests with cancellation overheads, IEEE/ACM Trans. Netw. 25 (2) (2017) 1279–1290.
[22] T. Hellemans, T. Bodas, B.V. Houdt, Performance analysis of workload dependent load balancing policies, Proc. ACM Meas. Anal. Comput. Syst. (PACM) 3 (2) (2019) 1–35.
[23] Z. Qiu, J.F. Pérez, R. Birke, L. Chen, P.G. Harrison, Cutting latency tail: Analyzing and validating replication without canceling, IEEE Trans. Parallel Distrib. Syst. 28 (11) (2017) 3128–3141.
[24] J.F. Pérez, L.Y. Chen, M. Villari, R. Ranjan, Holistic workload scaling: A new approach to compute acceleration in the cloud, IEEE Cloud Comput. 5 (1) (2018) 20–30.
[25] A. Vulimiri, O. Michel, P.B. Godfrey, S. Shenker, More is less: reducing latency via redundancy, in: ACM Works. Hot Topics in Netw. (HotNets), 2012, pp. 13–18.
[26] G. Ananthanarayanan, A. Ghodsi, S. Shenker, I. Stoica, Effective straggler mitigation: Attack of the clones, in: USENIX Symp. Netw. Sys. Design Implement. (NSDI), 2013, pp. 185–198.
[27] M. Primorac, K. Argyraki, E. Bugnion, When to hedge in interactive services, in: USENIX Symp. Netw. Sys. Design Implement. (NSDI), 2021, pp. 373–387.
[28] S. Liu, H. Xu, L. Liu, W. Bai, K. Chen, Z. Cai, RepNet: Cutting latency with flow replication in data center networks, IEEE Trans. Serv. Comput. 14 (1) (2021) 248–261.
[29] N.F. Maxemchuk, Dispersity routing in high-speed networks, Comput. Netw. ISDN Syst. 25 (6) (1993) 645–661.
[30] G. Joshi, Y. Liu, E. Soljanin, On the delay-storage trade-off in content download from coded distributed storage systems, IEEE J. Sel. Areas Commun. 32 (5) (2014) 989–997.
[31] G. Joshi, E. Soljanin, G. Wornell, Efficient redundancy techniques for latency reduction in cloud systems, ACM Trans. Model. Perform. Eval. Comput. Syst. (TOMPECS) 2 (2) (2017) 1–30.
[32] K. Lee, N.B. Shah, L. Huang, K. Ramchandran, The MDS queue: Analysing the latency performance of erasure codes, IEEE Trans. Inform. Theory 63 (5) (2017) 2822–2842.
[33] P. Parag, A. Bura, J.-F. Chamberland, Latency analysis for distributed storage, in: IEEE Inter. Conf. Comp. Commun. (INFOCOM), 2017, pp. 1–9.
[34] A. Badita, P. Parag, J.-F. Chamberland, Latency analysis for distributed coded storage systems, IEEE Trans. Inform. Theory 65 (8) (2019) 4683–4698.
[35] T. Hellemans, A. Yardi, T. Bodas, Download time analysis for distributed storage systems with node failures, in: IEEE Inter. Symp. Info. Theory (ISIT), 2021, pp. 2060–2065.
[36] R. Jinan, A. Badita, P. Sarvepalli, P. Parag, Low latency replication coded storage over memory-constrained servers, in: IEEE Inter. Symp. Info. Theory (ISIT), 2021, pp. 2340–2345.
[37] R. Jinan, A. Badita, P.K. Sarvepalli, P. Parag, Latency optimal storage and scheduling of replicated fragments for memory constrained servers, IEEE Trans. Inform. Theory 68 (6) (2022) 4135–4155.
[38] G. Joshi, D. Kaushal, Synergy via redundancy: Adaptive replication strategies and fundamental limits, IEEE/ACM Trans. Netw. 29 (2) (2021) 737–749.
[39] A. Badita, P. Parag, V. Aggarwal, Optimal server selection for straggler mitigation, IEEE/ACM Trans. Netw. 28 (2) (2020) 709–721.
[40] A. Badita, P. Parag, V. Aggarwal, Sequential addition of coded sub-tasks for straggler mitigation, in: IEEE Inter. Conf. Comp. Commun. (INFOCOM), 2020, pp. 746–755.
[41] A. Badita, P. Parag, V. Aggarwal, Single-forking of coded subtasks for straggler mitigation, IEEE/ACM Trans. Netw. 29 (6) (2021) 2413–2424.
[42] S. Lin, D.J. Costello, Error Control Coding: Fundamentals and Applications, second ed., Pearson Education India, 2011.
[43] M. Bramson, Y. Lu, B. Prabhakar, Randomized load balancing with general service time distributions, ACM SIGMETRICS Perform. Eval. Rev. 38 (1) (2010) 275–286.

[44] M. Bramson, Y. Lu, B. Prabhakar, Asymptotic independence of queues under randomized load balancing, Queueing Syst. 71 (3) (2012) 247–292.
[45] S. Shneer, A.L. Stolyar, Large-scale parallel server system with multi-component jobs, Queueing Syst. 98 (1) (2021) 21–48.
[46] T. Vasantam, On Occupancy Based Randomized Load Balancing for Large Systems with General Distributions (Ph.D. thesis), University of Waterloo, 2019.
[47] D. Cheng, J. Rao, Y. Guo, C. Jiang, X. Zhou, Improving performance of heterogeneous MapReduce clusters with adaptive task tuning, IEEE Trans. Parallel Distrib. Syst. 28 (3) (2017) 774–786.
[48] G. Koole, R. Righter, Resource allocation in grid computing, J. Sched. 11 (3) (2008) 163–173.
[49] R. Bitar, P. Parag, S.E. Rouayheb, Minimizing latency for secure distributed computing, in: IEEE Inter. Symp. Info. Theory (ISIT), 2017, pp. 2900–2904.
[50] A.O. Al-Abbasi, V. Aggarwal, Video streaming in distributed erasure-coded storage systems: Stall duration analysis, IEEE/ACM Trans. Netw. 26 (4) (2018) 1921–1932.
[51] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, K. Ramchandran, Speeding up distributed machine learning using codes, IEEE Trans. Inform. Theory 64 (3) (2018) 1514–1529.
[52] R. Bitar, P. Parag, S.E. Rouayheb, Minimizing latency for secure coded computing using secret sharing via staircase codes, IEEE Trans. Commun. 68 (8) (2020) 4609–4619.
[53] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, B.V. Houdt, A better model for job redundancy: Decoupling server slowdown and job size, IEEE/ACM Trans. Netw. 25 (6) (2017) 3353–3367.
[54] R. Bekker, S.C. Borst, O.J. Boxma, O. Kella, Queues with workload-dependent arrival and service rates, Queueing Syst. 46 (3) (2004) 537–556.
[55] A. Vulimiri, Latency-Bandwidth Tradeoffs in Internet Applications (Ph.D. thesis), University of Illinois at Urbana-Champaign, 2015.
[56] L. Liu, V.G. Kulkarni, Explicit solutions for the steady state distributions in M/PH/1 queues with workload dependent balking, Queueing Syst. 52 (4) (2006) 251–260.

Rooji Jinan received her Ph.D. degree from the Indian Institute of Science, Bengaluru in 2023, M. Tech. degree in communication engineering and signal processing in 2015, and B. Tech. degree in electronics and communication engineering in 2012, both from University of Calicut, Kerala. She worked as an assistant professor at Christ College of Engineering, Kerala, from 2016 to 2017. She is currently a Senior Engineer at Qualcomm India Private Ltd, Bangalore. Her research interests include channel modeling and network planning in advanced 5G networks, real time communication systems, and low latency distributed storage and compute systems.

Ajay Badita received the B.Tech. degree in electronics and communication engineering from SSCE, affiliated to JNTU Kakinada in 2011, the M.Tech. degree in electronics and communication engineering from NIT Rourkela in 2015, and the Ph.D. degree in electronics and communication engineering from the Indian Institute of Science, Bengaluru, in 2021. He was a Research Scientist at the IOTA Foundation, Berlin, between Aug 2021 and Aug 2022. His research interests include distributed ledgers, delay-sensitive communication, computation, and storage in distributed systems.

Tejas Bodas is currently an Assistant Professor in the Computer Systems Group at IIIT Hyderabad. Prior to this, he was a Scientist at TCS Research and an Assistant Professor at IIT Dharwad. He has also been a C. V. Raman postdoc at IISc, postdoc at LAAS, CNRS in Toulouse and a visiting postdoc at University of Antwerp. He received his Ph.D. in Electrical engineering from IIT Bombay. His research interests are in stochastic modeling, game theory, reinforcement learning and Bayesian optimization.

Parimal Parag received the B.Tech. and M.Tech. degrees in electrical engineering from IIT Madras in 2004 and the Ph.D. degree in electrical engineering from Texas A&M University in 2011. He joined the Indian Institute of Science in 2014, where he is currently an Associate Professor with the Department of Electrical Communication Engineering. Prior to that, he was a Senior System Engineer (Research and Development) with ASSIA Inc., Redwood City, CA, USA, from 2011 to 2014. His research interests include the design and analysis of large scale networked systems. He was a coauthor of the 2018 IEEE ISIT Student Best Paper. He was a recipient of the 2017 Early Career Award from the Science and Engineering Research Board.