HAL open science

Stability and Optimization of Speculative Queueing Networks

Jonatha Anselmi, Neil Walton

▶ To cite this version:

Jonatha Anselmi, Neil Walton. Stability and Optimization of Speculative Queueing Networks. IEEE/ACM Transactions on Networking, 2021, pp.1-12. 10.1109/TNET.2021.3128778 . hal-03349775

HAL Id: hal-03349775
https://hal.science/hal-03349775

Submitted on 20 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stability and Optimization of Speculative Queueing Networks

J. Anselmi[1] and N. Walton[2]

[1]: Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France.

[2]: University of Manchester, UK

jonatha.anselmi@inria.fr, neil.walton@manchester.ac.uk

Abstract

We provide a queueing-theoretic framework for job replication schemes based on the principle "*replicate a job as soon as the system detects it as a* straggler". This is called job *speculation*. Recent works have analyzed replication on arrival, which we refer to as *replication*. Replication is motivated by its implementation in Google's BigTable. However, systems such as Apache Spark and Hadoop MapReduce implement speculative job execution. The performance and optimization of speculative job execution is not well understood. To this end, we propose a queueing network model for load balancing where each server can speculate on the execution time of a job. Specifically, each job is initially assigned to a single server by a frontend dispatcher. Then, when its execution begins, the server sets a timeout. If the job completes before the timeout, it leaves the network, otherwise the job is terminated and relaunched or resumed at another server where it will complete. We provide a necessary and sufficient condition for the stability of speculative queueing networks with heterogeneous servers, general job sizes and scheduling disciplines. We find that speculation can increase the stability region of the network when compared with standard load balancing models and replication schemes. We provide general conditions under which timeouts increase the size of the stability region and derive a formula for the optimal speculation time, i.e., the timeout that minimizes the load induced through speculation. We compare speculation with redundant-d and redundant-to-idle-queue-d rules under an $S\&X$ model. For light loaded systems, redundancy schemes provide better response times. However, for moderate to heavy loadings, redundancy schemes can lose capacity and have markedly worse response times when compared with the proposed speculative scheme.

1 Introduction

Jobs taking longer than expected to complete, so-called *stragglers*, severely impact the performance of computer systems. Initial work on MapReduce note that stragglers have a significant impact on performance and occur for reasons that do not necessarily depend on the magnitude of the job's computational requirements but instead on the configuration and runtime of the underlying platform [21]. Causes for straggling jobs include run-time contention phenomena among CPU cores, processor caches, memory bandwidth, network bandwidth [48, 3]; bugs and errors in code [21]; unfortunate disk seek times or head locations [33]; or, energy requirements, power and temperature constraints, and maintenance activities [20]. All of which can cause a job to straggle significantly beyond its inherent computational requirement.

To mitigate the effect of stragglers on system-wide performance, researchers are currently proposing redundant, or hedged, requests [5, 20, 33, 45, 41, 4]. In this respect, there are two underlying principles: either replicate

"*replicate a job upon its arrival and use the results from whichever replica responds first*", [20];

or speculate

"*replicate a job as soon as the system detects it as a straggler*", [5].

The former is referred to as job replication, and here all redundant replicas are usually canceled as soon as one completes or starts service. The latter is referred to as job speculation [49], and here long jobs are usually killed after some time and replicated elsewhere. Clearly, the straggler detection monitor, or

timeout rule, needs to be smart enough to distinguish between jobs whose intrinsic size is large and jobs that are taking longer than expected to complete because of some unfortunate runtime phenomenon. There is increased interest in replication as it is employed by Google's BigTable, while the speculation principle is used in Apache Spark and Hadoop MapReduce.[1] For instance, it was reported in [41] that speculative tasks account for 25% of all tasks in Facebook's Hadoop cluster.

Both principles provide effective mitigation techniques for the straggler problem but which one is preferable is not currently clear. Replication may lead to significant additional resource costs while speculation may increase latency. One advantage of speculation is that only the long jobs are replicated, though this may come at the cost of an increased latency because a cloned job is executed after the system detected it as a straggler. Our objective is to provide the first theoretical framework for speculation in a stochastic and dynamic setting. We then use this to understand the impact of speculation and how to optimize its performance.

Recently, a number of theoretical works appeared in the literature for evaluating the performance of replication-based schemes in a stochastic and dynamic setting; see Section 2 for a review. These consider systems that replicate on arrival. Speculation, however, has received far less attention.

1.1 Model, Results and Contributions

In this paper, we develop the first theoretical framework for evaluating the delay performance achieved by the speculation principle in a queueing network. We consider a stochastic system of parallel queues, each with its own server. Upon arrival, each job is dispatched to a random queue and, when a job's service is first initiated, a timeout is set. If the timeout is reached and the job is not completed, then the job is killed and re-routed to another randomly selected queue where it must be completed.

Our first result, Theorem 1, characterizes the *stability* region induced by speculation. We show that speculative queueing networks are stable (positive Harris recurrent[2]) under the usual stability condition: that is, the nominal load at each queue is less than one. This result is non-trivial because: i) our network includes feedback mechanisms with mixture of different job size distributions; ii) we consider a general class of work-conserving scheduling disciplines and service time distributions; and iii) the stability region found differs from the one achieved by standard load balancing schemes such as join-the-shortest-queue. It is well known that under i) and ii) the "usual" stability condition may not be sufficient for positive recurrence [12, 42, 22].

Since timeouts induce a controllable constraint on the processing time of each job, we then consider the problem of optimal timeout design. First, we note that speculation can increase the system load and thus decrease the stability region, because part of the job is effectively served twice. Therefore, in Theorem 2, we derive criteria for speculation to increase the stability region and show that these criteria are satisfied in realistic scenarios. Then, we investigate optimal design and characterize the timeout that maximizes the size of the stability region in terms of a non-linear one-dimensional equation given in Theorem 4. This optimal timeout can be easily solved numerically and in some cases also analytically.

Overall, we use our results to investigate the advantage of speculation by performing two comparisons:

- First, we compare speculative and *standard* load balancing in the stationary regime; "standard" means no replication and no speculation, though dynamic information between servers and dispatcher(s) can be exchanged, as in power-of-d. We establish when it is advantageous to introduce timeouts as means to improve performance in an existing standard load balancing system. An immediate consequence of Theorem 1 is that a timeout increases the stability region if and only if the expected remaining time of the current execution is greater than the conditional expected service time at the second queue. We find large sets of timeouts having this property for important classes of service time distributions; e.g., Pareto, hyper-exponential and bimodal, see Section 5.1. We can then refine this analysis to give the optimal timeout for a given distribution in Theorem 4.

- Second, we compare speculation and replication in the stationary regime. This comparison is difficult to perform analytically because a satisfactory characterization of the stability region induced by replication is currently unknown even when considering homogeneous servers, outside special cases; see

[1] For instance, Apache Spark applies the attribute `spark.speculation` while Hadoop MapReduce uses task configuration `mapred.map.tasks.speculative.execution`.

[2] We point the reader unfamiliar with positive Harris recurrence to [15, Section 3]

[39, 37, 40, 6] and Section 2 for further details. However, we can provide reasoning and scenarios where speculation will improve response times when compared with replication. In particular, replication is bad when the shortest replica is stuck in queue, and instead service is completed by serving one or more larger replicas. The reasoning bears out in numerical simulations. Although replication provides better average response times in a light-load regime, which is to be expected, a large number of simulations indicate that speculative load balancing with the optimal timeout yields an increased stability region and short response times over a wide range of loads.

Our work is the first developing either of the above comparisons. Concisely, in terms of stability we find that speculative load balancing can be more advantageous than standard load balancing and replication.

1.2 Organization

The remainder of the paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we introduce speculative queueing networks. In Section 4, we present our stability theorem. In Section 5, we address the problem of designing optimal timeouts and compare the stability region achieved within our approach with standard load balancing and existing replication schemes. In Section 6, we present a conjecture about the mean waiting time obtained when servers are homogeneous. Finally, Section 7 draws the conclusions of this work and outlines further research.

2 Literature Review

Over the last decade there has been interest in the performance analysis of policies that dispatch to the shortest among a set of queues, for instance, power-of-two-choices and join-the-idle-queue; see [43] for a survey. However, more recently there has been an increased focus on job replication and in the following, we review existing replication approaches; for further details, we refer the reader to the recent survey [25].

Within replication, multiple replicas of a job are placed at different servers and redundant replicas are canceled, either when the first replica completes service (cancel-on-completion) or when the first replica initiates service (cancel-on-start). It is striking that both cancel-on-start and cancel-on-complete redundancy models are analytically tractable under independent exponentially distributed job sizes. This is first shown for cancel-on-complete in [26], where the reversibility results of the order-independent queue[9] are applied. The stationary distribution for cancel-on-start is found in [7], which applies results from [44]. The paper [35] is the first to analyze the latency and computing cost of the cancel-on-start and cancel-on-finish redundancy strategies. Subsequent works expand these results and thus our understanding of redundancy models [24]. The stationary distributions found are fully explicit; however, it is often hard to derive tractable formulas for key performance metrics. For this reason, existing works develop diffusion limits and mean field approximations for these models [14, 32]. Under i.i.d. exponential replica sizes, it is noted that there is neither loss nor gain of capacity due to redundancy job replication [11]. However, it is also noted in [23] that the independent exponential distributed job sizes is not a reasonable assumption, as job sizes will be correlated and slow service tends to result in heavy tailed service times. For this reason, works investigate the impact of replication under a variety of job size distributions and service disciplines [31]. The results in [39, 6] find that in general capacity is lost due to increased utilization of redundant jobs, though a full understanding of the stability region induced by cancel-on-completion schemes is currently unclear; see also [37].

Another similar approach is delayed replication. Here, each job is sent to an arbitrary server and if the server has not finished service of this job within some fixed time that includes both waiting and service, the job is replicated elsewhere. From the point of view of a single queue, this approach is somewhat equivalent to a queueing system with abandonments or reneging [27]. Delayed replication is investigated in [30] for homogeneous servers and under the assumption of "asymptotic independence", though the structure of the resulting stability region is not explicited.

While several recent works investigate replication, studies of speculation are largely empirical and analyze the performance and dependencies of specific platform implementations [5, 20, 33, 45, 41, 4, 41]. An early analysis of timeouts in queueing is given by [28], where all jobs initially join a single queue and then jobs that timeout are sequentially sent along a line of further queues. Some existing theoretical works focus on static settings with a finite number of jobs and no queueing [2, 46, 1].

The queueing and stability behavior of job speculation is not well-understood and we investigate its impact. We investigate stability using the fluid limit approach of Dai [15, 13]. We note that job sizes at different queues change due to both timeouts and correlations between jobs. In general, such networks can exhibit non-standard stability behavior; see, for instance, Bramson [12]. Although stability is non-trivial for these networks, we find that stability still holds when the nominal load is below one. This is important as the nominal load can be reduced through the use of speculation.

Further, we characterize the optimal timeout for a speculative load-balancing network. Again there are few works that investigate optimal timeouts, though we note that the paper [47] provide upper and lower bounds for a constrained utility optimization. This is then approximated for a timeout optimization under independent task-based service time distributions and in the presence of idle servers. In contrast, we consider a queueing framework under an S&X model where the optimal timeout minimizes load and thus maximizes stability. The use of Markov decision processes and optimal stopping theory [38] analysis appears to be somewhat new to this line of literature. We note the contemporaneous work of [34] consider a discrete time MDP formulation of optimal replication. This approach (along with associated implications for reinforcement learning and load balancing) would seem an important area of future investigation.

3 Speculative Queueing Network

We consider a network composed of N queues working in parallel. Queues have an infinite buffer, are work-conserving and are heterogeneous, in the sense that they process jobs at possibly different rates. Specifically, queue i processes jobs with rate $\mu_i > 0$, $i = 1, \ldots, N$, and without loss of generality we assume that $\sum_{i=1}^{N} \mu_i = d$. Jobs join the network via an exogenous Poisson process with rate λN and are initially dispatched to queue i independently with probability $p_{0,i}$. When a job starts its execution on its designated queue, say i, a clock is initialized and a *timeout*, τ_i, is set. If the timeout is reached and the job has not completed, then it is killed and re-routed independently to another queue where it is either resumed or re-executed from scratch. At this second queue, the job is then served until complete and leaves the network afterwards.

3.1 Dynamics and Assumptions

Let $\tau := (\tau_1, \ldots, \tau_N) \in \mathbb{R}_+^d$ denote the timeout vector. The n-th job that joins the network has associated the random variables: $\xi(n) \in \mathbb{R}_+$, the interarrival time between the n-th and $(n-1)$-th jobs; $\eta(n) := (\eta_1(n), \eta_2(n))$, the service requirements of job n where $\eta_j(n) \in \mathbb{R}_+$ represents the service time associated to its j-th execution if processed at unit rate; $\ell(n) := (\ell_1(n), \ell_2(n))$, where $\ell_j(n) \in \{1, \ldots, N\}$ denotes the queue associated to the j-th visit performed by job n; and $\tau(n) := \tau_{\ell_1(n)}$, the timeout of job n.

On the event

$$\frac{\eta_1(n)}{\mu_{\ell_1(n)}} \leq \tau(n),$$

job n completes service at $\ell_1(n)$ before the timeout $\tau(n)$ is reached and it leaves the network. In this case, the random variables $\eta_2(n)$ and $\ell_2(n)$ are not used. On the complementary event, job n is re-routed to queue $\ell_2(n)$ where it receives service for $\frac{\eta_2(n)}{\mu_{\ell_2(n)}}$ time units and leaves the network afterwards.

Within this notation, the primitive sequences driving the dynamics of the stochastic network under investigation are $\{\xi(n), n \geq 1\}$, $\{\eta(n), n \geq 1\}$ and $\{\ell(n), n \geq 1\}$, all defined on a common probability space. We assume that these sequences are i.i.d. and mutually independent. We note however that, for fixed n, $\eta_1(n)$ and $\eta_2(n)$ may have arbitrary dependency. For simplicity of notation, we will refer to $\eta_1(1)$ and $\eta_2(1)$ simply as η_1 and η_2, respectively. Because η_1 and η_2 may be dependent and have a possibly different distribution, this allows us to models two different scenarios:

- η_1 and η_2 are equal in distribution: upon re-routing, the job is killed and then re-started from scratch
- η_2 is stochastically smaller than η_1: upon re-routing, the job is stopped and then resumed.

The case of identical replicas, i.e., $\eta_1 = \eta_2$ (ω-per-ω) is permitted but it should be clear that in this case speculation does not help because the amount of work to be done does not change after re-routing. We

remark that speculation is motivated by the fact that something can go wrong on the servers' runtimes, i.e., a given job may have different service times if executed on different machines. This rules out the case of identical replicas. We also require that $\mathbb{E}[\eta_1(n)+\eta_2(n)] < \infty$ and that $\mathbb{P}(\frac{\eta_1}{\mu_i} \leq \tau_i) > 0$ for all i, which means that jobs complete and have a chance of completing before the timeout.

We assume that each $\xi(n)$ is exponentially distributed with mean $(\lambda N)^{-1}$, so that arrivals occur according to a Poisson process of rate λN. For all $i, j \in \{1, \ldots, N\}$, let $p_{i,j} := \mathbb{P}(\ell_2(n) = j \mid \ell_1(n) = i)$. We assume that $p_{i,j} = p_{1,j}$ for all i, i.e., the internal routing probabilities depend on the destination j but not on the source i. This assumption makes all jobs 'homogeneous in terms of re-routing' and is required in the proof of Theorem 2; without this assumption, the proposed queueing network may be unstable even under the usual stability condition, as in [12].

Since queues speculate on service times, we refer to the proposed model as a *speculative queueing network*.

3.2 Scheduling Disciplines

Each queue operates under a work-conserving head-of-the-line scheduling discipline. We recall that a service discipline is head-of-the-line if within each class at the queue jobs are served in order their arrival [13]. This includes First-come first-served (FCFS), Head-of-the-line processor-sharing (HLPS) or class-based priority disciplines. On the other hand, Processor-Sharing (PS) is excluded, though we believe that the results presented in this paper apply by a separate argument [36, Section 3.3]; due to page constraints, we do not discuss this. In our framework, the class of a job refers to the queue that the job is at and whether the job has been timed-out or not; see Section 4.1 for further details. With a class-based priority discipline, jobs are ranked according to their class. The higher the rank, the higher the priority. Depending on whether the discipline is preemptive or not, the job in execution may be stopped if a job with higher priority arrives. For instance, within a given queue, a job that has been re-routed may be given higher (or lower) priority than job that have currently visited only one queue.

We observe that some scheduling disciplines may process more than one job at a time. In this case, the clock associated to the execution of each job does not increase linearly over time, but in relation to the processing capacity devoted to the execution of the job itself.

3.3 Further Notation

We use the convention that products (resp. sums) over empty sets are one (resp. zero). The set of non-negative real numbers is denoted by $\mathbb{R}_+$. The indicator function of A is denoted by $\mathbb{I}_A$. We use $|\cdot|$ to denote set cardinality and $\|\cdot\|$ to denote the L_1 norm. We let $p_0 := (p_{0,i} : i = 1, ..., N)$ and $p_1 := (p_{1,i} : i = 1, ..., N)$ denote the routing probability vectors upon job arrival in the network and re-routing, respectively. Unless specified otherwise, indices i and j will range over $\{1, \ldots, N\}$. For $a, b \in \mathbb{R}$, $(a)^+ = \max\{a, 0\}$, $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. For a differentiable function $t \mapsto f(t)$, $f'(t)$ denotes its derivative in t.

4 Stability

In this section, we first provide a multiclass representation of the speculative queueing network introduced in previous section and show that this representation belongs to the family of queueing networks investigated in [15]. This will allow us to use Theorem 4.2 of [15], which provides a criterion to establish the stability (positive Harris recurrence) of the Markov process, say X, describing the dynamics of the network under investigation. This criterion is expressed in terms of an associated *fluid model*, which we define below.

4.1 Multiclass Representation

We now provide a multiclass representation of our queueing network described in Section 3.1 where classes are used to distinguish between jobs that have timeout or not. The advantage of this is that we can directly represent jobs within the framework of Dai [15] and Bramson [13]. Thus we can analyze our fluid model as a fluid limit as expressed within that framework.

Let us consider the set of classes $\mathcal{K} := \mathcal{K}_1 \cup \mathcal{K}_2$ where $\mathcal{K}_1 := \{(k,i) : k \in \{c,u\}, 1 \leq i \leq N\}$ and $\mathcal{K}_2 := \{(i,j) : 1 \leq i,j \leq N\}$ represent the set of exogenous and endogenous classes, respectively. Further, we use 'c' for jobs that will complete service at their first queue, and 'u' for jobs that will receive uncompleted service at their first queue (and thus will timeout). On the same probability space used to define the speculative queueing network above, we now construct a queueing network composed of such classes.

Jobs enter the network through class $(c,i) \in \mathcal{K}_1$ or $(u,i) \in \mathcal{K}_1$ for $i = 1, ..., N$. These jobs join queue i and, after processing at i, a class-(c,i) job leaves the network, while a class-(u,i) job is re-routed and joins queue j as a class $(i,j) \in \mathcal{K}_2$ job with probability $p_{i,j}$.

The inter-arrival times of jobs in class $(c,i) \in \mathcal{K}_1$ follow a Poisson process that is obtained by thinning the Poisson process associated to $\{\xi(n), n \geq 1\}$ with respect to the Bernoulli process $\{B_n^{c,i}, n \geq 1\}$ where

$$B_n^{c,i} := \mathbb{I}_{\left\{\ell_1(n)=i,\, \frac{\eta_1(n)}{\mu_i} \leq \tau_i\right\}}.$$

Under the Poisson thinning (aka splitting) property, the set of service times of class-(c,i) jobs can be equivalently obtained by sampling independently from the distribution of $(\frac{\eta_1}{\mu_i} | \frac{\eta_1}{\mu_i} \leq \tau_i)$. Similarly, the interarrival times of jobs in class $(u,i) \in \mathcal{K}_1$ follow a Poisson process that is obtained by thinning the Poisson process associated to $\{\xi(n), n \geq 1\}$ with respect to the Bernoulli process $\{B_n^{u,i}, n \geq 1\}$ where

$$B_n^{u,i} := \mathbb{I}_{\left\{\ell_1(n)=i,\, \frac{\eta_1(n)}{\mu_i} > \tau_i\right\}}$$

and they have deterministic service times equal to τ_i.

Under the Poisson thinning property, the set of service times of class-(i,j) jobs can be equivalently obtained by sampling independently from the distribution of $(\frac{\eta_2}{\mu_j} | \frac{\eta_1}{\mu_i} > \tau_i)$.

The dynamics (routing decisions, arrival and service times) of the multiclass network above are equivalent to the dynamics of the speculative queueing network introduced in Section 3. Equivalences of this type are commonly applied in the analysis of queueing networks; for instance, see Section 2.7 of [16].

4.2 Markov and Fluid Model

We now define a continuous-time Markov process $X(t)$ that describes the dynamics of the multiclass queueing network described above. Specifically, taken to be right continuous, consider the pair

$$X(t) := (\mathbb{Q}(t), V(t)). \tag{1}$$

Here, $\mathbb{Q}(t) = (\mathbb{Q}_1(t), \ldots, \mathbb{Q}_N(t))$ and

$$\mathbb{Q}_i(t) = (k_{i,1}, k_{i,2}, ..., k_{i,Q_i(t)})$$

where $k_{i,n} \in \mathcal{K}$ gives the class of the n-th job in queue i and $Q_i(t)$ is the total number of jobs at queue i. This captures how jobs are lined up in queue i. Also $V(t) = (V_k(t) : k \in \mathcal{K})$ with $V_k(t)$ denoting the remaining service time of the class-k job in execution and with the convention that $V_k(t) = 0$ if such job does not exist.

The process $X = \{X(t), t \geq 0\}$, living on state space $\mathcal{X}$, is a piecewise deterministic Markov process (PDMP) and satisfies the strong Markov property; see page 362 in [19]. If X is positive Harris recurrent, we recall that a unique ergodic stationary distribution π exists and

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t Q_i(s)\, \mathrm{d}s = \int_{\mathcal{X}} q_i(x)\, \pi(\mathrm{d}x) \tag{2}$$

almost surely, for any initial configuration, where $Q_i(t)$ is the queue length of server i at time t and $q_i(x)$ is the queue length of server i for network state $x \in \mathcal{X}$.

Now, we define a fluid model for the dynamics of the queueing network under investigation. For $t \in \mathbb{R}_+$, let $\overline{Q}(t) = (\overline{Q}_k(t), k \in \mathcal{K})$, $\overline{T}(t) = (\overline{T}_k(t), k \in \mathcal{K})$ and $\overline{I}(t) = (\overline{I}_i(t), i = 1, \ldots, N)$ and consider the following conditions

$$\overline{Q}_{c,i}(t) = \overline{Q}_{c,i}(0) + \lambda dp_{0,i}\mathbb{P}(\eta_1 \leq \tau_i\mu_i)\, t - \frac{(\overline{T}_{c,i}(t) - \overline{V}_{c,i})^+}{\mathbb{E}[\frac{\eta_1}{\mu_i} \mid \frac{\eta_1}{\mu_i} \leq \tau_i]} \geq 0 \tag{3}$$

$$\overline{Q}_{u,i}(t) = \overline{Q}_{u,i}(0) + \lambda dp_{0,i}\mathbb{P}(\eta_1 > \tau_i\mu_i)\, t - \frac{(\overline{T}_{u,i}(t) - \overline{V}_{u,i})^+}{\tau_i} \geq 0 \tag{4}$$

$$\overline{Q}_{j,i}(t) = \overline{Q}_{j,i}(0) - \frac{(\overline{T}_{j,i}(t) - \overline{V}_{j,i})^+}{\mathbb{E}[\frac{\eta_2}{\mu_i} \mid \frac{\eta_1}{\mu_j} > \tau_j]} + \frac{p_{j,i}}{\tau_j}(\overline{T}_{u,j}(t) - \overline{V}_{u,j})^+ \geq 0, \quad \forall (j,i) \in \mathcal{K}_2 \tag{5}$$

$$\overline{T}_k(t) \text{ is nondecreasing and starts from zero}, \quad \forall k \in \mathcal{K} \tag{6}$$

$$\overline{I}_i(t) = t - T_{c,i}(t) - \bar{T}_{u,i}(t) - \sum_{j \neq i} \overline{T}_{j,i}(t) \text{ is nondecreasing} \tag{7}$$

$$\int_0^\infty \left(\overline{Q}_{c,i}(t) + \overline{Q}_{u,i}(t) + \sum_{j \neq i} \overline{Q}_{j,i}(t) \right) \mathrm{d}\overline{I}_i(t) = 0 \tag{8}$$

for all $i \in \{1, \ldots, N\}$ where $(x)^+ = \max(x, 0)$ and $\overline{V} \in \mathbb{R}_+^{|\mathcal{K}|}$ is interpreted as the remaining service time vector. These conditions define our *fluid solutions*. A *fluid solution* is any solution $(\overline{Q}(\cdot), \overline{T}(\cdot))$ to (3)-(8). The quantities in (3)-(7) have the following interpretation: at the fluid scale, $\overline{Q}_k(t)$ is the amount of jobs of class k at time t, $\overline{T}_k(t)$ is the cumulative time dedicated to the processing of class k jobs by time t, and $\overline{I}_i(t)$ is the cumulative idle time of queue i by time t. For instance, Equation (4) says that the fluid amount of class-(u,i) jobs increases with rate $\lambda dp_{0,i}\mathbb{P}(\eta_1 > \tau_i\mu_i)$ due to external arrivals and decreases by $\frac{(\overline{T}_{u,i}(t) - \overline{V}_{u,i})^+}{\tau_i}$ as these jobs stay in queue i for τ_i time units. Similar interpretations are easily obtained for $\overline{Q}_{c,i}(t)$ and $\overline{Q}_{j,i}(t)$. Equation (8) is therefore interpreted as the work-conserving condition.

Remark 1. *The set of fluid solutions associated to a speculative queueing network is identical to the set of fluid solutions associated to its corresponding multiclass queueing network where the routing and service processes are independent; see Formulas (4.17)-(4.21) in [15].*

From the above remark and from Theorem 4.1 of [15], it follows that the fluid model equations are the limit equations satisfied by a speculative queueing network. We refer the reader to Theorem 4.1 for a formal statement and proof of this fluid limit argument, from which our stated fluid model follows as a direct consequence.

We now define stability for the *fluid model*.

Definition 1. *We say that the fluid model is* stable *if there exists a constant $\delta > 0$ such that for any fluid solution with $\|\overline{Q}(0)\| + \|\overline{U}\| + \|\overline{V}\| = 1$, it holds that $\overline{Q}(t + \delta) = 0$ for all $t > 0$.*

The constant δ above may depend on all model parameters but not on the initial state.

The following theorem connects fluid model stability with $X(t)$ being positive Harris recurrent and was first proved in [15].

Proposition 1 (Dai [15]; Bramson [13])**.** *Assume that* (3)-(8) *hold. If the fluid model solutions is stable, then $X(t)$ is positive Harris recurrent.*

The proposition follows from Theorem 4.16 of Bramson [13]. In the Appendix, we give a proof that essentially explains how our model fits within the framework in [13]. We note that there are some aspects of our model that are non-standard within the usual fluid stability framework: i) jobs are routed according to their size rather than according to an independent mechanism, and ii) the size of jobs within a queue can have a different probability distribution (jobs having non-identical distributions within a multi-class queue can impact stability, see the well-known counter-example of Bramson [12]). However, in our case, it is possible to segment arrivals into an extended set of job classes as described in Section 4.1. This yields the required independence properties for routing probabilities for jobs within each class. This deals with part i) above. For part ii), we note that fluid stability remains sufficient for positive recurrence so long as we have a Markovian state description and head-of-the-line service within each class. Each of these apply to our model and therefore what remains is to prove the stability of the fluid model. This result is stated below.

4.3 Fluid Stability and Positive Harris Recurrence

We now state our main result on stability. Let

$$\rho_i := \lambda N p_{0,i} \mathbb{E}\left[\frac{\eta_1}{\mu_i} \wedge \tau_i\right] + \lambda N \sum_{j=1}^{N} p_{0,j} p_{1,i} \mathbb{P}(\eta_1 > \mu_j \tau_j) \mathbb{E}\left[\frac{\eta_2}{\mu_i} \,\middle|\, \frac{\eta_1}{\mu_j} > \tau_j\right] \tag{9}$$

be the *nominal load* of queue i, for all i. This accounts for the work from both speculative and non-speculative jobs.

The following result shows that the Markov process of interest is stable under the condition that $\rho_i < 1$ for all i. Our proof is based on Proposition 1 and on a Lyapunov argument; see the Appendix.

Theorem 1. *For all i, assume that $\rho_i < 1$. Then, X is positive Harris recurrent.*

Within the proposed queueing network, the mean service time of a job depends on the number of visits currently performed in the network and that the network topology is not feedforward. In these cases, it is well known that the usual stability condition is in general not sufficient to make the underlying Markov process positive Harris recurrent. A counterexample with Poisson arrivals, exponentially distributed service times and two FCFS queues is the reentrant line developed in [12]; see also [42, 22].

We note that the stability condition in Theorem 1 differs from the "usual" condition $\lambda N p_{0,i}/\mu_i < 1$, which is the nominal stability condition when no timeout is set (no speculation). It is thus natural to investigate which approach yields the largest stability region. This is one of the objectives of the following section.

5 Timeout Design

In this section, we rely on Theorem 1 to perform some optimization and understand the advantages of speculative load balancing. Specifically, since timeouts and routing probabilities are under the control of the network manager, we investigate whether or not it is possible to design them in a manner such that the resulting stability region is larger than the stability region achieved by

i) standard load balancing; i.e., no replication and no speculation, though servers and dispatcher(s) can exchange dynamic control messages as in join-the-shortest-queue;

ii) other replication schemes based on the principle "replicate upon job arrivals".

Remark 2. *We will address the question above when servers are homogeneous, that is $\mu_i = 1$ for all i. The heterogeneous case can be handled as well but at the cost of complicating the exposition unnecessarily. For this reason, we limit this application of Theorem 1 to the homogeneous case only.*

Given $\tau \in \mathbb{R}_+$, let us define

$$\rho(\tau) := \lambda \mathbb{E}[\eta_1 \wedge \tau] + \lambda \mathbb{P}(\eta_1 > \tau) \mathbb{E}[\eta_2 \mid \eta_1 > \tau] \tag{10}$$

and note that $\rho(\tau) = \rho_i$ if the routing probabilities are identical and all timeouts are equal to τ.

The following proposition says that if servers are homogeneous, then it is optimal to choose identical timeouts and routing probabilities; see the Appendix for a proof.

Proposition 2. *If $\mu_i = 1$ for all i, then*

$$\inf_{\tau \in \mathbb{R}_+} \rho(\tau) = \inf \max_i \rho_i \tag{11}$$

where $\rho(\tau)$ is defined in (10) *and the second* inf *is taken over all stochastic routing vectors $p_0, p_1 \in \mathbb{R}_+^N$ and timeout vectors $\tau \in \mathbb{R}_+^N$.*

In view of Proposition 2, in the analysis that follows we will require the following assumption, which implies that the speculative queueing network is symmetric.

Assumption 1 (Symmetric network)**.** *For all i, $\mu_i = 1$, $p_{0,i} = p_{1,i} = \frac{1}{N}$ and $\tau_i = \tau$, for some $\tau \in \mathbb{R}_+$.*

In the following, we first investigate scenarios where timeouts help to increase the stability region with respect to standard load balancing. Then, we characterize an optimal timeout and provide numerical simulations to compare the performance of speculation with existing replication schemes.

5.1 Speculation vs Standard Load Balancing

Under Assumption 1, we investigate when the proposed approach yields an increased stability region with respect to standard load balancing. In the latter case, it is well known that $\lambda\mathbb{E}[\eta_1] < 1$ is the necessary and sufficient stability condition for the Markov process that models the dynamics induced by several dispatching algorithms such as join-the-shortest-queue. Therefore, given Theorem 1, we aim at finding $\tau \in \mathbb{R}_+$ such that

$$\rho(\tau) < \lambda\mathbb{E}[\eta_1]. \tag{12}$$

This problem can be easily addressed at least numerically once probability distributions for the η_i's are known. Nonetheless, in this section our aim is to find insights and general conditions ensuring that (12) holds.

We have the first general condition; see the Appendix for a proof.

Theorem 2. *Let Assumption 1 hold. Then, $\rho(\tau) < \lambda\mathbb{E}[\eta_1]$ if and only if*

$$\mathbb{E}[\eta_2 \mid \eta_1 > \tau] < \mathbb{E}[\eta_1 - \tau \mid \eta_1 > \tau]. \tag{13}$$

The inequality (13) admits a simple interpretation because the RHS term is the expected remaining service time of the job in progress after age τ and the LHS term is the expected service time of a second execution if the job would timeout at τ.

We now derive a further condition introducing some structure on the service time random variables η_i, $i = 1, 2$. In the following, S, S_1, S_2 and X are four auxiliary nonnegative random variables each independent of all else and such that S, S_1 and S_2 are equal in distribution.

Assumption 2 (The "$S\&X$" model)**.** *For $i = 1, 2$, $\eta_i = S_i X$.*

The service time structure in Assumption 2 was first introduced in [23]: X is interpreted as a job's *intrinsic size* and S_i represents the *server slowdown* incurred by a job's i-th execution. An important special case is obtained when X is deterministic, which models systems where the service time variability is only due to server runtime phenomena. Here, the service times associated to the execution of a job are independent; a line of papers investigate this case, e.g., [24, 40].

Remark 3. *Assumption 2 models the case where jobs are re-executed from scratch after re-routing and rules out the possibility that they may be resumed. This is a worst case in our respect because speculation (and our model) allows jobs to be resumed after re-routing, and the stability region obtained by resuming jobs is clearly increased.*

We will also make use of the following assumption.

Assumption 3. *For some $z \in \mathbb{R}_+$,*

$$\mathbb{E}\left[Sx \wedge z\right] < \mathbb{P}\left(Sx \leq z\right) \mathbb{E}[S]\, x, \quad \forall x \in \text{support}(X). \tag{14}$$

The following result, proven in the Appendix, provides a sufficient condition for which speculation yields an increased stability region with respect to standard load balancing.

Theorem 3. *Let Assumptions 1 and 2 hold. Then, $\rho(z) < \lambda\mathbb{E}[\eta_1]$ for all $z \in \mathbb{R}_+$ that satisfy Assumption 3. Furthermore, if X is deterministic, then Assumption 3 is also necessary to have $\rho(z) < \lambda\mathbb{E}[\eta_1]$.*

Theorem 3 shows that Assumption 3 is almost the requirement for speculation to increase the size of the stability region and we now briefly discuss it. Towards this purpose, let $\mathcal{T}_{S,X}$ be the set of $z \in \mathbb{R}_+$ such that (14) holds. Assuming that $X = 1$, it is not difficult to show that:

1. If S is s_m with probability p and $s_M > s_m$ with probability $1 - p$ (a bimodal distribution), then $\mathcal{T}_{S,X} = (s_m, (s_M - s_m)p)$; typically, $s_M \gg s_m$ and $p \approx 0.99$ [8, Section 2.6.4], and therefore $\mathcal{T}_{S,X}$ is not empty in practice.

2. If S follows a nondegenerate hyperexponential distribution, then $\mathcal{T}_{S,X} = (0, \infty)$;

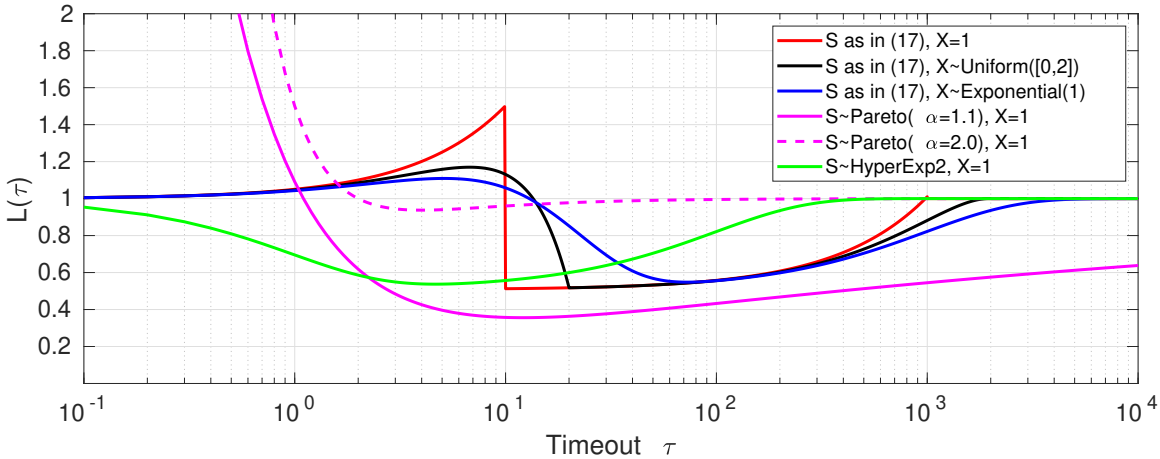

Figure 1: Speculative vs standard load balancing via $L(\tau)$, (15), under a number of $S\&X$ models.

3. If $S \sim \text{Pareto}(\alpha, s_m)$, where $\alpha > 1$ and s_m respectively denote the shape and scale parameters, then $\mathcal{T}_{S,X} = (\alpha s_m, \infty)$.

On the other hand, if $X = 1$ and S follows the "more deterministic" Erlang distribution, then one can show that $\mathcal{T}_{S,X}$ is empty and in this case Theorem 3 implies that speculation loses capacity.

Now, let

$$L(\tau) := \frac{\rho(\tau)}{\lambda \mathbb{E}[\eta_1]}. \tag{15}$$

If $L(\tau) < 1$, then $L(\tau)$ can be interpreted as the *load reduction* with respect to standard load balancing when adopting timeout τ. As in Section 2.6.4 of [8], let us assume that

$$S = \begin{cases} 10 & \text{w.p. } 0.99 \\ 10^3 & \text{w.p. } 0.01, \end{cases} \tag{16}$$

i.e., S follows a bimodal distribution. By increasing the timeout τ, Figure 1 plots $L(\tau)$ under a number of combinations on the distributions of S and X. For the Pareto distribution, we have chosen $s_m = 1$ and $\alpha \in \{1.1, 1.5\}$; it is known that $\alpha \in [1, 1.5]$ is the most common range [4, 3, 29]. For the hyperexponential distribution, we have chosen two phases with balanced means, as in [29]. Specifically, the probability and rate vectors are $[0.99, 0.01]$ and $[1, 1/99]$, respectively. If timeouts are too small, Figure 1 shows that speculation may loose capacity ($L(\tau) > 1$), and this is to be expected because jobs have almost no chance of completing before the timeout. If timeouts are large enough, however, they increase the size of the stability region ($L(\tau) < 1$). We notice that the timeouts that minimize the load can double the size of the stability region achieved by standard load balancing because within such timeouts $L \approx 0.5$. When S has the bimodal form given in (16), this benefit is obtained whenever $\tau \in [10, 10^3)$ (if $X = 1$), $\tau \in [13, 2 \times 10^3)$ (if $X \sim$ Uniform([0,2])), and $\tau \in [13.5, \infty)$ (if $X \sim$ Exponential(1)). When S is Pareto, we observe that $L(\tau)$ decreases when α decreases for any given τ, i.e., timeouts are particularly helpful when the service time variability increases. When S is hyperexponential, we also observe that all timeouts reduces the load. This is in agreement with the statement in point 2) above.

Since the set of τ's ensuring that $L(\tau) < 1$ is somewhat large, we conclude that replication by speculation is robust with respect to small errors in service time distributions and thus that the system manager has some flexibility when designing a proper timeout rule.

5.2 An Optimal Timeout

Our stability result, Theorem 1, demonstrates that the load, $\rho(\tau)$, is the key first order performance indicator for a speculative network. Theorems 2 and 3 show that timeouts can reduce load and thus increase capacity.

Now, we investigate the minimizers of the load function $\tau \mapsto \rho(\tau)$, i.e., we characterize the optimal timeout. We address this problem under the following assumption.

Assumption 4. *We assume that η_1 has a probability density function $f_1(t)$ with a decreasing hazard function, i.e.,*

$$t \mapsto \frac{f_1(t)}{\int_t^\infty f_1(s)ds} \tag{17}$$

is decreasing. Further, we assume that $\bar{\eta}_2(t) := \mathbb{E}[\eta_2|\eta_1 > t]$ is such that

$$t \mapsto \frac{1 + \bar{\eta}_2'(t)}{\bar{\eta}_2(t)} \tag{18}$$

is non-decreasing.

The function (17) is the hazard function of a probability distribution and it is well-known that a decreasing hazard function is a characteristic of a heavy-tailed probability distribution. A trivial condition for (18) to be non-decreasing is that η_2 is independent of η_1. This corresponds to the case where job sizes are fixed but variability in service occurs due to independent random events at each server. In this case, we note that (17) states that the complementary CDF of η_1 is log-convex. This complements with the findings of [35] where it is found that independent log-convex distributions service time distributions benefit from cancel-on-complete replication.

Let us define the optimal timeout as follows.

Definition 2 (Optimal Timeout)**.** *The optimal time τ is the smallest time such that*

$$\frac{f_1(\tau)}{\int_\tau^\infty f_1(s)ds} \leq \frac{1 + \bar{\eta}_2'(\tau)}{\bar{\eta}_2(\tau)}\,. \tag{19}$$

If we assume that η_2 is independent of η_1, then we notice that the optimal timeout rule reduces to the condition

$$\frac{f_1(\tau)}{\int_\tau^\infty f_1(s)ds} = \frac{1}{\mathbb{E}[\eta_2]}. \tag{20}$$

Definition 2 provides a practical rule for speculation because it allows one to apply reinforcement learning or statistical estimation techniques to learn the optimal timeout when the service time distributions are not known in advance.

The following is our main result on optimal timeout design; see the Appendix for a proof.

Theorem 4. *Let Assumptions 1 and 4 hold. Finite optimal timeouts (see Definition 2) exist and minimize the load $\rho(\tau)$ induced on a speculative queueing network. Moreover, if, in addition, the service requirements η_1 and η_2 are independent, then any value of τ satisfying* (20) *minimizes the load.*

5.3 Speculation vs Replication

We now compare the performance achieved by Speculative Load Balancing (SLB) with the performance achieved by strategies that replicate jobs at the time of their arrival in the network. We perform such comparison within the $S\&X$ model in Assumption 2, that is a worst case scenario for speculation in view of Remark 3.

First, let us consider a setting where each incoming job is replicated to d queues selected uniformly at random and independently of anything else. Here, redundant replicas are canceled as soon as one either completes (Cancel-on-Complete) or starts (Cancel-on-Start) service. We refer to the former resp. latter scheme as CoC-d resp. CoS-d.

It is known that CoS-d is equivalent to the Least-Left-Workload-d (LLW-d) dispatching algorithm. We recall that LLW-d sends each incoming job to a queue having the shortest remaining workload among d selected uniformly at random, with ties broken randomly. Since stability for CoS-d is obtained if and only if $\lambda\mathbb{E}[\eta_1] < 1$, i.e., as in standard load balancing, we have the following remark.

Remark 4. *SLB yields an increased stability region w.r.t. CoS-d whenever the conditions in Theorems 2 and 3 are satisfied.*

On the other hand, it is difficult to perform a comparison with CoC-d because a satisfactory characterization of the resulting stability region is currently unclear [39, 37, 40]. However, depending on the load regime, we can argue intuitively as follows:

- Assume the network is *lightly loaded*, or over-provisioned. Then, the time spent in the system by a job with CoC-d is $\min(\eta_1, \ldots, \eta_d)$, where η_c represents the service time of copy $c = 1, \ldots, d$. This should be compared to $\eta_1 \wedge \tau + \eta_2 \mathbb{I}_{\{\eta_1 > \tau\}}$, the time spent by a job in a speculative queueing network within the same load condition. For any $d \geq 2$, it is not difficult to show that[3] $\min(\eta_1, \ldots, \eta_d) \leq \eta_1 \wedge \tau + \eta_2 \mathbb{I}_{\{\eta_1 > \tau\}}$, which means that, in a lightly loaded regime, it is better to replicate upon job arrivals rather than speculate upon straggler detection. This claim should also be intuitive.
- Assume the network is *moderately loaded*. Then, within CoC-d it may happen that the first copy that completes is not the minimum and that the fast copy gets stuck in its queue. For instance, for a given job n, let us assume that $\eta_2(n) \geq \eta_1(n)$ and that the copy that completes first when applying CoC-d is the one of size $\eta_2(n)$. On the event $\eta_1(n) \leq \tau$, speculation on that job induces a lower load and no price is paid for the extra copies.
- Assume the network is *heavily loaded*. When λ increases, the scenario depicted in the moderate load regime amplifies, potentially leading CoC-d to be unstable. Here, it is not intuitive which approach is better than the other.

To support the intuition above, we present the results obtained by running a large set of numerical simulations. In our tests, we assume the service time structure given in Assumption 2. Figure 2 plots the average response time (time spent in the system) obtained within SLB, CoC-d and CoS-d, for $d = 2, 4$ by increasing $\lambda\mathbb{E}[\eta_1]$ while keeping $\mathbb{E}[\eta_1]$ constant and under a number of $S\&X$ models where $\mathbb{E}[X] = 1$ and the random variable S is given by (16) (bimodal distribution) or follows a Pareto distribution with shape parameter $\alpha \in \{1.1, 1.5, 2\}$ over the support $[1, 10^3]$. For SLB, we have chosen the timeouts τ^* specified in the respective subfigures. Each point $(*)$ in each curve refers to an average of 50 simulations and each simulation executes 10^7 jobs. We also assume $N = 50$ FCFS queues.

The plots in Figure 2 confirm the intuition above. CoC-d provides the best results in light load conditions (as expected) and it also increases the stability region achieved by CoS-d, that is the stability region of standard join-the-shortest-queue-like algorithms ($\lambda\mathbb{E}[\eta_1] < 1$). However, in most cases, SLB goes further and is able to accommodate more traffic than CoC-d. For the most extreme heavy tailed distribution, Pareto with $\alpha = 1.1$, CoC-2 eventually provides the best results, though within CoC-d is not clear how to well choose d *a priori*.

We now compare SLB with Redundant-to-Idle-Queue-d (RIQ-d), a replication scheme that works as CoC-d but replicas are only made to those servers which are idle, and if no idle server is found then the job is sent to a random one of the d selected; see [23]. RIQ-d was introduced to avoid the potential loss of capacity of CoC-d, though at the same time we observe that it can ban its potential gain. We also notice that comparing SLB and RIQ-d is not completely fair because the latter scheme is *dynamic* in the sense that it needs dispatcher(s) and queues to continuously exchange information about servers' status. No feedback mechanisms to dispatcher(s) are assumed in SLB, though they could clearly be integrated to further reduce delays; we do not discuss this in this paper. Within the same setting described above, Figure 2 plots the average response time obtained within SLB and RIQ-d, for $d = 2, 4$ by increasing $\lambda\mathbb{E}[\eta_1]$ while keeping $\mathbb{E}[\eta_1]$ constant. RIQ-4 provides slightly better results in light load conditions, which is to be expected because as $\lambda \downarrow 0$ it behaves as CoC-4. As the load increases, however, it is less and less likely to find idle queues and the dynamics of RIQ-d get closer and closer to the dynamics of Random (RND), which sends each job to a single server selected independently at random. For the latter, the stability condition $\lambda\mathbb{E}[\eta_1] < 1$ applies, while SLB preserves stability much further.

Finally, we conclude this section with the following remark about the communication overhead induced by SLB, CoC-d, CoS-d and RIQ-d. SLB requires $1 + \mathbb{P}(\eta_1 > \tau) < 2$ control messages per job in average. On the other hand, CoC-d and CoS-d require, for each job, d dispatch messages plus $d-1$ cancellation messages.

[3] We recall that η_1 and η_2 are assumed equal in distribution in this section.

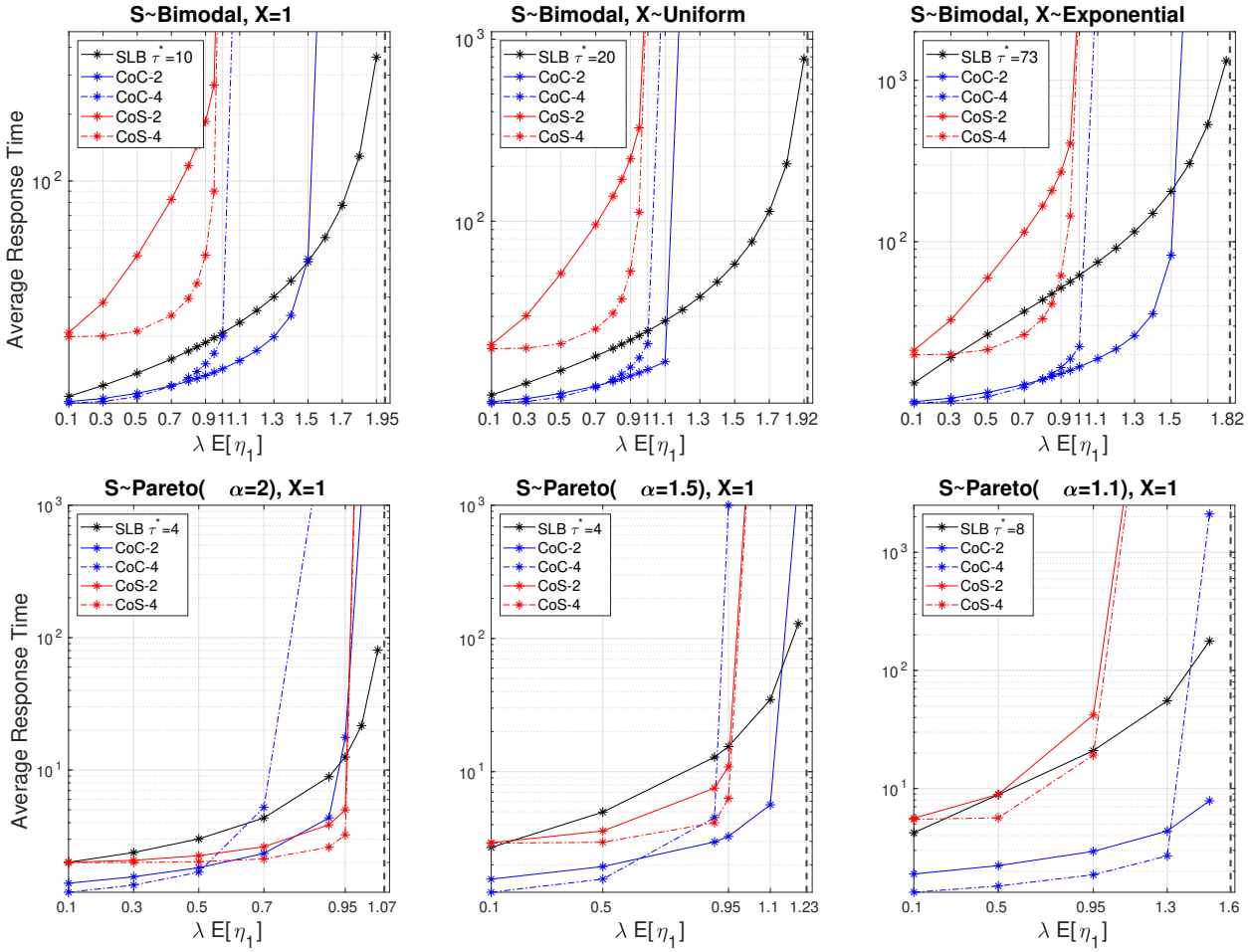

Figure 2: Average response time obtained within Speculative Load Balancing (SLB), Cancel-on-Complete-d (CoC-d) and Cancel-on-Start-d (CoS-d) under $S\&X$ models. The vertical dashed black lines represent the limits of the stability region of SLB.

6 Response Time for Large Systems

In this section, we consider a symmetric (see Assumption 1) speculative queueing network composed of FCFS queues. Within these assumptions, let $R_{N,n}(\tau)$ be the overall time spent by the n-arriving job in the system and let

$$R_N(\tau) := \limsup_{m\to\infty} \frac{1}{m}\sum_{n=1}^{m} \mathbb{E}[R_{N,n}] \tag{21}$$

be the average *response time* experienced by jobs. In this section, our goal is to investigate $R_N(\tau)$.

First, we observe that $R_N(\infty)$ corresponds to the mean response time of an M/GI/1 queue. When $\tau < \infty$, the feedback speculation mechanism significantly complicates the analysis and our aim is to develop an approximation. We focus on the large system limiting regime where $N \to \infty$ and $\rho(\tau)$ is constant. Defining

$$W := \frac{\lambda}{2}(1 + \mathbb{P}(\eta_1 \geq \tau))\frac{M}{1-\rho(\tau)} \tag{22}$$

$$M := \frac{\mathbb{E}[(\eta_1 \wedge \tau)^2] + \mathbb{E}[\hat{\eta}_2^2]\,\mathbb{P}(\eta_1 > \tau)}{1 + \mathbb{P}(\eta_1 > \tau)} \tag{23}$$

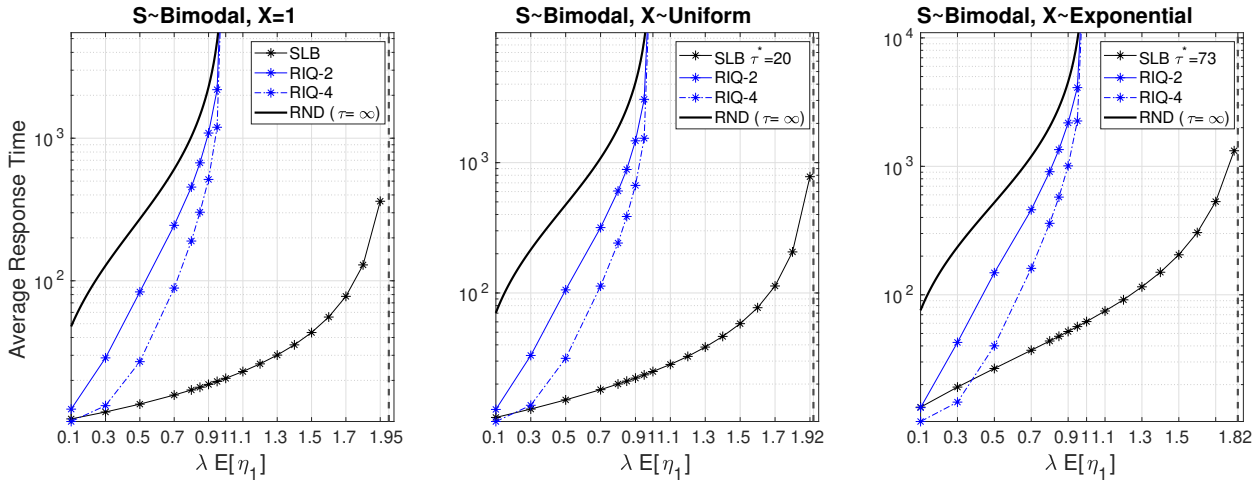

Figure 3: Average response time obtained within Speculative Load Balancing (SLB) and Redundant-to-Idle-Queue-d (RIQ-d) under $S\&X$ models. The vertical dashed black lines represent the limits of the stability region of SLB.

where $\hat{\eta}_2$ is an auxiliary random variable equal in distribution to $\eta_2 \mid \eta_1 > \tau$, we claim that the following conjecture holds true.

Conjecture 1. *Provided that* $\rho(\tau) < 1$,

$$\lim_{N\to\infty} R_N(\tau) = (1 + \mathbb{P}(\eta_1 > \tau))W + \frac{\rho(\tau)}{\lambda}. \tag{24}$$

The underlying justification behind Conjecture 1 lies in postulating that queues become "asymptotically independent" in the limit $N \to \infty$. In this case, the arrival process at each queue i is the superposition of an exogenous rate-λ Poisson process and N independent feedback processes each with rate $\lambda\,\mathbb{P}(\eta_1 > \tau)/N$. Given that the intensity of each of the feeback processes approaches zero as $N \to \infty$, we may assume that their superposition is "approximately" a Poisson process with rate $\lambda\,\mathbb{P}(\eta_1 > \tau)$, provided that N is "large". This is justified by the Palm–Khinchine theorem, which ensures that the superposition of several independent sparse point processes converges weakly to a Poisson process on $\mathbb{R}_+$; see [18, Chapter 5.8]. Therefore, we conjecture that the arrival process at each queue i is the independent superposition of 1) a rate-λ Poisson process carrying jobs of sizes equal in distribution to η_1 and 2) a rate-$\lambda\,\mathbb{P}(\eta_1 > \tau)$ Poisson process carrying jobs of sizes equal in distribution to $\eta_2|\eta_1 \geq \tau$. This queue can be interpreted as an M/G/1 queue where the arrival rate is $\lambda + \lambda\,\mathbb{P}(\eta_1 > \tau)$ and the service times are equal in distribution to H, where

$$H = \begin{cases} \overline{\eta}_1 \wedge \tau & \text{w.p. } \frac{1}{1+\mathbb{P}(\eta_1>\tau)} \\ \eta_2 \mid \eta_1 \geq \tau & \text{otherwise} \end{cases} \tag{25}$$

with $\overline{\eta}_1$ equal in distribution to η_1 but independent of all else. Note that $\mathbb{E}[H^2] = M$. Furthermore, the traffic intensity at this M/G/1 queue is $(\lambda + \lambda\mathbb{P}(\eta_1 > \tau))\mathbb{E}[H] = \rho(\tau)$ and the mean workload observed at arrival times is W, which follows by applying the Pollaczek-Khinchine formula. Conditioning on $\eta_1 \leq \tau$, the mean response time of a job is

$$W + \mathbb{E}[\eta_1 \mid \eta_1 \leq \tau], \tag{26}$$

and conditioning on $\eta_1 > \tau$, the mean response time of a job is

$$W + \tau + W + \mathbb{E}[\eta_1 \mid \eta_1 \leq \tau], \tag{27}$$

that is the time spent during the first visit, $W + \tau$, plus the time spent during the second visit, $W + \mathbb{E}[\eta_1 \mid \eta_1 \leq \tau]$. Putting (26) and (27) together, we get

$$\mathbb{P}(\eta_1 \leq \tau)(W + \mathbb{E}[\eta_1 \mid \eta_1 \leq \tau]) + \mathbb{P}(\eta_1 > \tau)(2W + \tau + \mathbb{E}[\eta_2|\eta_1 \geq \tau]),$$

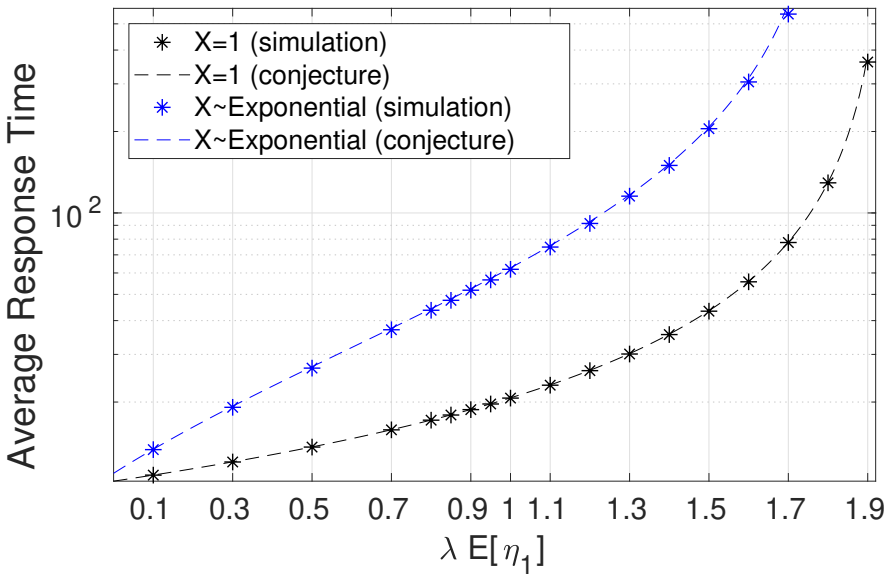

Figure 4: Average response times obtained by simulation and via Conjecture 1; S has the form (16) and $N = 50$.

which after some algebra boils down to (24).

Figure 4 plots the average time spent by a job in a speculative queueing network obtained by simulation ($N = 50$) and by the conjectured Formula (24). Each point ($*$) of each plot refers to an average of 50 simulations and each simulation was based on 10^7 jobs. As in the plots of Section 5.3, we assume $\eta_1 = S_1 X$ and $\eta_2 = S_2 X$ where S is as in (16). For X, we distinguish the cases $X = 1$ and $X \sim$ Exponential(1). It is remarkable how the approximation is accurate under any load condition.

7 Conclusions

In this paper, we provide a first performance evaluation for job speculation in a queueing network. We characterize the stability region of a speculative queueing network and find the resulting stability region to be distinct from a queueing network implementing a standard load balancing scheme. We provide conditions on job size distributions for speculative load balancing to increase the size of the stability region. Specifically, in the presence of heavy tailed server slowdown, we find that speculation is a good mechanism to increase stability and thus throughput. We provide the first characterization of the optimal timeout for a speculative task. This follows simple implementable formulas (19) and, under independence, (20). Under moderate to heavy loadings, simulations indicate that speculation can significantly improve performance when compared with redundancy schemes and standard load-balancing systems. Finally, using the Pollaczek-Khinchine formula, we postulate on the impact of speculation on response times by providing a numerically accurate approximation.

The proposed framework opens up important questions:

- It is possible to design schemes that combine the benefits of replication for a light loaded regime, while maintaining the desirable stability properties of speculation for moderate to high loads.
- The design of the optimal timeout requires the solution of a Markov decision process. Since task sizes must be statistically estimated, real systems may need to apply reinforcement learning to design optimal task based replication decisions.
- The analysis in Section 6 forms a conjecture on the mean field limit of a speculative queueing network. The resolution of this conjecture requires further analysis.

- We have applied random server assignment to arriving and speculative tasks. However, it may be preferable to implement join-idle-queue or join-the-shortest-queue-d routing to these tasks. While we believe that the stability region obtained within such speculative dynamic schemes remains unchanged, delays are expected to be further reduced.

A natural generalization of our model considers multiple speculation levels. Specifically, timeouts can be used in second job executions and a job that timeouts in the second visited queue gets routed to a third queue, and so forth a number of times. We believe that our results generalize to this setting naturally, though a formal proof for Theorem 1 would require more work.

Although job speculation is widely used in practice, its performance has been understudied relative to recent works on redundancy. The proposed framework sheds a new light on speculative load balancing and addresses a number of important questions on the design of job replication in large scale computer systems.

Acknowledgment

This work is supported by the French National Research Agency in the framework of the "Investissements d'avenir" program (ANR-15-IDEX-02) and the LabEx PERSYVAL (ANR-11-LABX-0025-01).

References

[1] M. F. Aktas, P. Peng, and E. Soljanin. Straggler mitigation by delayed relaunch of tasks. *SIGMETRICS Perform. Eval. Rev.*, 45(3):224231, Mar. 2018.

[2] M. F. Akta and E. Soljanin. Straggler mitigation at scale. *IEEE/ACM Transactions on Networking*, 27(6):2266–2279, Dec 2019.

[3] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. Why let resources idle? aggressive cloning of jobs with dolly. In *Proc. of the 4th USENIX Conference on Hot Topics in Cloud Computing*, HotCloud12, page 17, USA, 2012. USENIX Association.

[4] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. Effective straggler mitigation: Attack of the clones. In *Proc. of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, page 185198, USA, 2013. USENIX Association.

[5] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris. Reining in the outliers in map-reduce clusters using mantri. In *Proc. of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, page 265278, USA, 2010. USENIX Association.

[6] E. Anton, U. Ayesta, M. Jonckheere, and I. M. Verloop. On the stability of redundancy models. *arXiv preprint arXiv:1903.04414*, 2019.

[7] U. Ayesta. On redundancy-d with cancel-on-start aka join-shortest-work (d). *ACM SIGMETRICS Performance Evaluation Review*, 46(2):24–26, 2019.

[8] L. A. Barroso and U. Hoelzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 1st edition, 2009.

[9] S. Berezner, C. Kriel, and A. E. Krzesinski. Quasi-reversible multiclass queues with order independent departure rates. *Queueing Systems*, 19(4):345–359, 1995.

[10] D. P. Bertsekas. Dynamic programming and optimal control 3rd edition, volume ii. *Belmont, MA: Athena Scientific*, 2011.

[11] T. Bonald and C. Comte. Balanced fair resource sharing in computer clusters. *Performance Evaluation*, 116:70–83, 2017.

[12] M. Bramson. Instability of fifo queueing networks. *Ann. Appl. Probab.*, 4(2):414–431, 05 1994.

[13] M. Bramson. *Stability of queueing networks*. Springer, 2008.

[14] E. Cardinaels, S. Borst, and J. S. van Leeuwaarden. Redundancy scheduling with locally stable compatibility graphs. *arXiv preprint arXiv:2005.14566*, 2020.

[15] J. G. Dai. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability*, pages 49–77, 1995.

[16] J. G. Dai and J. M. Harrison. *Processing Networks: Fluid Models and Stability*. Cambridge University Press, 2020.

[17] J. G. Dai and G. Weiss. Stability and instability of fluid models for reentrant lines. *Math. Oper. Res.*, 21(1):115134, Feb. 1996.

[18] M. J. S. Daniel P. Heyman. *Stochastic Models in Operations Research: Stochastic Processes and Operating Characteristics*. Dover, 2003.

[19] M. H. A. Davis. Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):353–376, 1984.

[20] J. Dean and L. A. Barroso. The tail at scale. *Commun. ACM*, 56(2):7480, Feb. 2013.

[21] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[22] D. Gamarnik and J. J. Hasenbein. Instability in stochastic and fluid queueing networks. *Ann. Appl. Probab.*, 15(3):1652–1690, 08 2005.

[23] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, and B. V. Houdt. A better model for job redundancy: Decoupling server slowdown and job size. *IEEE/ACM Transactions on Networking*, 25(06):3353–3367, nov 2017.

[24] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, M. Velednitsky, and S. Zbarsky. Redundancy-d: The power of d choices for redundancy. *Operations Research*, 65(4):1078–1094, 2017.

[25] K. Gardner and R. Righter. Product forms for fcfs queueing models with arbitrary server-job compatibilities: An overview. *arXiv preprint arXiv:2006.05979*, 2020.

[26] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytiä, and A. Scheller-Wolf. Queueing with redundant requests: exact analysis. *Queueing Systems*, 83(3-4):227–259, 2016.

[27] F. A. Haight. Queueing with reneging. *Metrika*, 2(1):186–197, 1959.

[28] M. Harchol-Balter. Task assignment with unknown duration. *J. ACM*, 49(2):260–288, 2002.

[29] M. Harchol-Balter, A. Scheller-Wolf, and A. R. Young. Surprising results on task assignment in server farms with high-variability workloads. SIGMETRICS '09, pages 287–298, New York, NY, USA, 2009. ACM.

[30] T. Hellemans, T. Bodas, and B. Van Houdt. Performance analysis of workload dependent load balancing policies. *SIGMETRICS Perform. Eval. Rev.*, 47(1):78, Dec. 2019.

[31] T. Hellemans and B. V. Houdt. Performance of redundancy (d) with identical/independent replicas. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 4(2):1–28, 2019.

[32] T. Hellemans and B. Van Houdt. Heavy traffic analysis of the mean response time for load balancing policies in the mean field regime. *arXiv preprint arXiv:2004.00876*, 2020.

[33] H. Huang, W. Hung, and K. G. Shin. Fs2: Dynamic data replication in free disk space for improving disk performance and energy consumption. *SIGOPS Oper. Syst. Rev.*, 39(5):263276, Oct. 2005.

[34] G. Joshi and D. Kaushal. Synergy via redundancy: Adaptive replication strategies and fundamental limits. *arXiv preprint arXiv:2012.13608*, 2020.

[35] G. Joshi, E. Soljanin, and G. Wornell. Efficient redundancy techniques for latency reduction in cloud systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2(2):1–30, 2017.

[36] F. Kelly. *Reversibility and Stochastic Networks*. Wiley, Chicester, 1979.

[37] G. Mendelson. A lower bound on the stability region of redundancy-d with fifo service discipline. *Operations Research Letters*, 49(1):113 – 120, 2021.

[38] G. Peskir and A. Shiryaev. *Optimal stopping and free-boundary problems*. Springer, 2006.

[39] Y. Raaijmakers and S. Borst. Achievable stability in redundancy systems. *Proc. ACM Meas. Anal. Comput. Syst.*, 4(3), Nov. 2020.

[40] Y. Raaijmakers, S. Borst, and O. Boxma. Redundancy scheduling with scaled bernoulli service requirements. *Queueing Systems*, 93(1):67–82, Oct 2019.

[41] X. Ren, G. Ananthanarayanan, A. Wierman, and M. Yu. Hopper: Decentralized speculation-aware cluster scheduling at scale. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 379392, New York, NY, USA, 2015. Association for Computing Machinery.

[42] T. I. Seidman. "first come, first served" can be unstable! *IEEE Transactions on Automatic Control*, 39(10):2166–2171, Oct 1994.

[43] M. van der Boor, S. C. Borst, J. S. van Leeuwaarden, and D. Mukherjee. Scalable load balancing in networked systems: A survey of recent advances. *arXiv preprint arXiv:1806.05444*, 2018.

[44] J. Visschers, I. Adan, and G. Weiss. A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Systems*, 70(3):269–298, 2012.

[45] A. Vulimiri, P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker. Low latency via redundancy. In *Proc. of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pages 283–294, New York, NY, USA, 2013. ACM.

[46] D. Wang, G. Joshi, and G. W. Wornell. Efficient straggler replication in large-scale parallel computing. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 4(2), Apr. 2019.

[47] H. Xu and W. C. Lau. Optimization for speculative execution in big data processing clusters. *IEEE Transactions on Parallel and Distributed Systems*, 28(2):530–545, 2016.

[48] Y. Xu, M. Bailey, B. Noble, and F. Jahanian. Small is better: Avoiding latency traps in virtualized data centers. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, New York, NY, USA, 2013. Association for Computing Machinery.

[49] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica. Improving mapreduce performance in heterogeneous environments. In *Osdi*, volume 8, page 7, 2008.

Proof of Proposition 1

The proof below identifies how our model fits within the framework of Bramson [13], under the multi-class structure given in Section 4.1.

Proof of Proposition 1. The proof follows from arguments in Bramson [13]. Here we confirm that conditions stated in [13] apply to our model and we refer the reader to appropriate sections and results in that work.

We note that, typically in the fluid analysis of queueing networks, it is assumed that the arrival, service and routing processes are independent. This is not true in our case under our first state description in Section 3. However, under the extended class structure in Section 4.1, the state description does have independent job sizes (within each class). Further it should be noted that notions of positive Harris recurrence and petite sets remain without these independence assumptions (see the remark following [15, Proposition 2.1]). First we note that the process is a (Harris) Markov process. In particular, the state descriptor given in (1) and given by Bramson [13, Section 4.1 & 4.3] remains Markov in our setting, since when we condition on this state descriptor all future events are function of independent random variables. Second, we assume head-of-the-line service. These conditions ensure that closed sets are petite; see [13, Proposition 4.8].

Finally, we observe that the conditions of [13, Theorem 4.16] have now been met. Thus as concluded in the theorem, queueing network stability (positive Harris recurrence) holds whenever the associated fluid model is stable. □

Proof of Theorem 1

In view of Lemma 5.3 in [15], we can restrict to fluid solutions where $\overline{V} = 0$. Therefore, we can assume that for all i

$$\overline{Q}_{c,i}(t) = \overline{Q}_{c,i}(0) + \lambda N p_{0,i}\mathbb{P}(\eta_1 \le \tau_i\mu_i)t - \frac{\overline{T}_{c,i}(t)}{\mathbb{E}[\frac{\eta_1}{\mu_i} \mid \frac{\eta_1}{\mu_i} \le \tau_i]} \ge 0 \tag{28}$$

$$\overline{Q}_{u,i}(t) = \overline{Q}_{u,i}(0) + \lambda N p_{0,i}\mathbb{P}(\eta_1 > \tau_i\mu_i)t - \frac{\overline{T}_{u,i}(t)}{\tau_i} \ge 0 \tag{29}$$

$$\overline{Q}_{j,i}(t) = \overline{Q}_{j,i}(0) - \frac{\overline{T}_{j,i}(t)}{\mathbb{E}[\frac{\eta_2}{\mu_i} \mid \frac{\eta_1}{\mu_j} > \tau_j]} + p_{1,i}\,\frac{\overline{T}_{u,j}(t)}{\tau_j} \ge 0, \forall (j,i) \in \mathcal{K}_2. \tag{30}$$

For all i, let $G_i(t)$ be such that

$$G_i(t)\frac{p_{1,i}}{\mu_i} = \mathbb{E}\left[\frac{\eta_1}{\mu_i} \mid \frac{\eta_1}{\mu_i} \le \tau_i\right]\overline{Q}_{c,i}(t) + \tau_i\,\overline{Q}_{u,i}(t) + \sum_j \mathbb{E}\left[\frac{\eta_2}{\mu_i} \mid \frac{\eta_1}{\mu_j} > \tau_j\right]\left(p_{1,i}\overline{Q}_{u,j}(t) + \overline{Q}_{j,i}(t)\right) \tag{31}$$

and note that $G_i(t)$ is absolutely continuous. The function $G_i(t)$ may be interpreted as a Lyapunov function on queue i and we note that it does not correspond to the workload in queue i. Substituting (28)-(30) in (31), after some algebra we obtain

$$\begin{aligned}
G_i(t)\frac{p_{1,i}}{\mu_i} &= G_i(0)\frac{p_{1,i}}{\mu_i} + \lambda N p_{0,i}\mathbb{E}\left[\frac{\eta_1}{\mu_i} \wedge \tau_i\right]t - \overline{T}_{c,i}(t) - \overline{T}_{u,i}(t) \\
&\quad + \sum_j \mathbb{E}\left[\frac{\eta_2}{\mu_i} \mid \frac{\eta_1}{\mu_j} > \tau_j\right] p_{1,i}\lambda N p_{0,j}\mathbb{P}(\eta_1 > \tau_j\mu_j)t - \overline{T}_{j,i}(t) \\
&= \frac{G_i(0)}{\mu_i} - \overline{B}_i(t) + \lambda N p_{0,i}\mathbb{E}\left[\frac{\eta_1}{\mu_i} \wedge \tau_i\right]t + \sum_j \mathbb{E}\left[\frac{\eta_2}{\mu_i} \mid \frac{\eta_1}{\mu_j} > \tau_j\right] p_{1,i}\lambda N p_{0,j}\mathbb{P}(\eta_1 > \tau_j\mu_j)\,t \\
&= \frac{G_i(0)}{\mu_i} - \overline{B}_i(t) + \rho_i\,t
\end{aligned}$$

where $\overline{B}_i(t) := \overline{T}_{c,i}(t) + \overline{T}_{u,i}(t) + \sum_{j\neq i}\overline{T}_{j,i}(t)$ is interpreted as the cumulative time queue i has been busy (non-idle) in $[0,t]$. Now, assuming that t is a point of differentiability for $G_i(t)$

$$\dot{G}_i(t)\frac{p_{1,i}}{\mu_i} = -\mathbb{I}_{\{\mathcal{Q}_i(t)>0\}} + \rho_i \tag{32}$$

where $\mathcal{Q}_i(t) := \overline{Q}_{c,i}(t) + \overline{Q}_{u,i}(t) + \sum_{j\neq i}\overline{Q}_{j,i}(t)$ is the total number of jobs in queue i at time t. Thus, whenever $\mathcal{Q}_i(t) > 0$,

$$\dot{G}_i(t)\frac{p_{1,i}}{\mu_i} = \rho_i - 1 < 0. \tag{33}$$

On the other hand, if $\mathcal{Q}_i(t) = 0$, $\mathcal{Q}_s(t) > 0$ and $s \neq i$, then

$$
\begin{aligned}
G_i(t) - G_s(t) &= \left(\sum_{j \neq i} \mathbb{E}\left[\eta_2 \mid \frac{\eta_1}{\mu_j} > \tau_j \right] \overline{Q}_{u,j}(t) \right) - G_s(t) \\
&\leq \left(\sum_{j \neq i} \mathbb{E}\left[\eta_2 \mid \frac{\eta_1}{\mu_j} > \tau_j \right] \overline{Q}_{u,j}(t) \right) - \mathbb{E}\left[\frac{\eta_1}{p_{1,s}} \mid \frac{\eta_1}{\mu_s} \leq \tau_s \right] \overline{Q}_{c,s}(t) - \frac{\tau_s \mu_s}{p_{1,s}} \overline{Q}_{u,s}(t) \\
&\quad - \sum_{j \neq i} \mathbb{E}\left[\eta_2 \mid \frac{\eta_1}{\mu_j} > \tau_j \right] \left(\overline{Q}_{u,j}(t) + \frac{\overline{Q}_{j,s}(t)}{p_{1,s}} \right) \\
&\leq -\mathbb{E}\left[\frac{\eta_1}{p_{1,s}} \mid \frac{\eta_1}{\mu_s} \leq \tau_s \right] \overline{Q}_{c,s}(t) - \frac{\tau_s \mu_s}{p_{1,s}} \overline{Q}_{u,s}(t) - \sum_{j \neq i} \mathbb{E}\left[\eta_2 \mid \frac{\eta_1}{\mu_j} > \tau_j \right] \frac{\overline{Q}_{j,s}(t)}{p_{1,s}} < 0
\end{aligned}
$$

where the last inequality follows because $\mathcal{Q}_s(t) > 0$. Since s is generic, we obtain

$$
G_i(t) < \min_{s \neq i} G_s(t) \tag{34}
$$

provided that $\mathcal{Q}_i(t) = 0$.

Let $G(t) := \max_{i=1,\ldots,N} G_i(t)$. With (33) holding true when $\mathcal{Q}_i(t) > 0$ and (34) holding true when $\mathcal{Q}_i(t) = 0$ and $Q_s(t) > 0$, we can apply Lemma 3.2 of [17], which implies that $G(t)$ is also an absolutely continuous nonnegative function and such that $\dot{G}(t) < -\epsilon$ for some $\epsilon > 0$, provided that t is a point of differentiability of $G, G_1, \ldots, G_d$ such that $G(t) > 0$. Applying Lemma 2.2 of [17], we obtain that $G(t) = 0$ for all $t \geq \delta$, for some $\delta > 0$. Finally, we notice that $G(t) = 0$ implies that the system is empty.

Proof of Proposition 2

We have

$$
\begin{aligned}
\frac{1}{\lambda} \max_{i=1,\ldots,N} \rho_i &= \max_{i=1,\ldots,N} N p_{0,i} \mathbb{E}\left[\eta_1 \wedge \tau_i \right] + N p_{1,i} \sum_j p_{0,j} \mathbb{P}(\eta_1 > \tau_j) \mathbb{E}\left[\eta_2 \mid \eta_1 > \tau_j \right] \\
&\geq \sum_i p_{0,i} \left(\mathbb{E}\left[\eta_1 \wedge \tau_i \right] + \mathbb{P}(\eta_1 > \tau_i) \mathbb{E}\left[\eta_2 \mid \eta_1 > \tau_i \right] \right) \\
&\geq \min_i \mathbb{E}\left[\eta_1 \wedge \tau_i \right] + \mathbb{P}(\eta_1 > \tau_i) \mathbb{E}\left[\eta_2 \mid \eta_1 > \tau_i \right] \\
&\geq \inf_{t \in \mathbb{R}_+} \left(\mathbb{E}\left[\eta_1 \wedge t \right] + \mathbb{P}(\eta_1 > t) \mathbb{E}\left[\eta_2 \mid \eta_1 > t \right] \right) = \inf_{t \in \mathbb{R}_+} \frac{\rho(t)}{\lambda}
\end{aligned}
$$

as desired.

Proof of Theorem 2

Given $\tau \in \mathbb{R}_+$, we notice that

$$
\begin{aligned}
\frac{\rho(\tau)}{\lambda} - \mathbb{E}[\eta_1] &= \mathbb{E}[\eta_1 \wedge \tau] + \mathbb{P}(\eta_1 > \tau) \mathbb{E}[\eta_2 \mid \eta_1 > \tau] - \mathbb{E}[\eta_1] \\
&= \tau\, \mathbb{P}(\eta_1 > \tau) + \mathbb{P}(\eta_1 > \tau) \mathbb{E}[\eta_2 \mid \eta_1 > \tau] - \mathbb{E}[\eta_1 \mid \eta_1 > \tau] \mathbb{P}(\eta_1 > \tau) < 0
\end{aligned}
$$

if and only if (13) holds.

Proof of Theorem 3

Using Assumption 2 and that X, S_1 and S_2 are independent, we obtain

$$\begin{aligned}
\frac{\rho(\tau)}{\lambda} &= \mathbb{E}[(S_1X)\wedge\tau] + \mathbb{P}(S_1X>\tau)\,\mathbb{E}[S_2X \mid S_1X>\tau] \\
&= \sum_x \mathbb{E}[(S_1X)\wedge\tau \mid X=x]\,\mathbb{P}(X=x) + \mathbb{E}[S_2XI_{\{S_1X>\tau\}}] \\
&= \sum_x \mathbb{E}[(S_1x)\wedge\tau]\,\mathbb{P}(X=x) + \sum_x \mathbb{E}[S_2xI_{\{S_1x>\tau\}}|X=x]\mathbb{P}(X=x) \\
&= \sum_x \mathbb{E}\left[S_1x\wedge\tau\right]\,\mathbb{P}(X=x) + \sum_x \mathbb{E}[S_2x]\,\mathbb{P}(X=x)\mathbb{P}(S_1x>\tau) \\
&= \sum_x \mathbb{P}(X=x)\left(\mathbb{E}\left[S_1x\wedge\tau\right] + \mathbb{E}[S_2x]\mathbb{P}(S_1x>\tau)\right).
\end{aligned}$$

Since $\mathbb{E}[\eta_1] = \mathbb{E}[X]\mathbb{E}[S]$, we notice that

$$\mathbb{E}[\eta_1] - \frac{\rho(\tau)}{\lambda} = \sum_x \mathbb{P}(X=x)\Big(\mathbb{E}[Sx]\,\mathbb{P}(Sx\le\tau) - \mathbb{E}\left[Sx\wedge\tau\right]\Big) \tag{35}$$

and within Assumption 3 it is clear that the RHS of (35) is strictly greater than zero when $\tau = z$, for some $z \in \mathbb{R}_+$. The final part of the proposition is straightforward.

Optimal Stopping Formulation of Theorem 4

First, we explain how the proof of Theorem 4 can be expressed as an optimal stopping problem; in the following, stopping time and timeout are used interchangeably. Then, we show how the optimal timeout can be verified as an application of the one-step-lookahead principle. We also give a proof in continuous time and with deterministic stopping times. The more general proof (continuous-time stochastic timeouts) is more involved and, due to space constraints, is included in the appendix.

If we wish to minimize the time spent by a job in the processing phase, then we must minimize

$$V = \min_{\tau\in\mathcal{T}} \mathbb{E}\left[\eta_1\wedge\tau + \eta_2\mathbb{I}_{\{\tau\le\eta_1\}}\right] \tag{36}$$

$$= \min_{\tau\in\mathcal{T}} \mathbb{E}\left[\int_0^\tau \bar{F}_1(t)dt + \bar{\eta}_2(\tau)\bar{F}_1(\tau)\right] \tag{37}$$

where $\mathcal{T}$ is the set of stopping times on $\mathbb{R}_+$, the functions $f_1(t)$ and $\bar{F}_1(t)$ are respectively the pdf and ccdf of η_1, and $\bar{\eta}_2(t) := \mathbb{E}[\eta_2 \mid \eta_1 \ge t]$. The minimization (36) is an optimal stopping problem with continuation cost $\bar{F}_1(t)$ and stopping cost $\bar{\eta}_2(t)\bar{F}_1(t)$. We note that the objective function above is equal to $\rho(\tau)/\lambda$ where $\rho(\tau)$ is the induced load on the speculative queueing network for timeout τ. Thus, we aim at finding the stopping time that minimizes the load.

Discrete Time Argument. What follows is a brief informal argument for why the stopping condition (19) is correct. If we discretize time as $\mathcal{T}_\Delta := \{0, \Delta, 2\Delta, ...\}$ and if we are only allowed to stop on this restricted set of times (rather than $\mathcal{T}$), then the optimization for V above is a discrete time Markov decision process. The Bellman equation for this problem is

$$V(t) = \min\left\{p_\Delta(t)[\Delta + V(t+\Delta)], \bar{\eta}_2(t)\right\},$$

where we define $p_\Delta(t) := \mathbb{P}(\eta_1 \ge t+\Delta|\eta_1 \ge t) = 1 - \frac{f_1(t)\Delta}{\bar{F}_1(t)} + o(\Delta)$.

The one-step-look-ahead is known to be optimal for a wide class of optimal stopping rules [10, Section 4.4]. Here, we stop (or timeout) if it is better to stop now than continue one time step and then stop. In our case, it is easy to see that this corresponds to the condition:

$$\bar{\eta}_2(t) \le p_\Delta(t)[\Delta + \bar{\eta}_2(t+\Delta)].$$

The RHS term can be simplified:

$$
\begin{aligned}
&p_\Delta(t)[\Delta + \bar{\eta}_2(t+\Delta)] \\
&= \left(1 - \frac{f_1(t)\Delta}{\bar{F}_1(t)}\right)[\Delta + \Delta\bar{\eta}_2'(t) + \bar{\eta}_2(t)] + o(\Delta) \\
&= \bar{\eta}_2(t) + \Delta\left[1 + \bar{\eta}_2'(t) - \frac{f_1(t)}{\bar{F}_1(t)}\bar{\eta}_2(t)\right] + o(\Delta)\,.
\end{aligned}
$$

Observing the term in square brackets above, we see that up to terms of order $o(\Delta)$ the one-step look-ahead rule gives the condition to stop at τ whenever

$$
\frac{1 + \bar{\eta}_2'(\tau)}{\bar{\eta}_2(\tau)} \geq \frac{f_1(\tau)}{\bar{F}_1(\tau)}\,.
$$

This gives the stopping rule stated in (19). For one-step look-ahead to be optimal, we require that this set is closed, meaning that whenever the stopping condition is satisfied it remains satisfied for all future times. This is the motivation for Assumption 4.

Continuous Time Argument. If we restrict ourselves to deterministic stopping times, then it is clear that τ is the optimal stopping time: by conditioning on the value of η_1, we have that the objective (36) satisfies

$$
\mathbb{E}\left[\eta_1 \wedge \tau + \eta_2 \mathbb{I}_{\{\tau \leq \eta_1\}}\right] = \int_0^\tau \bar{F}_1(t)dt + \bar{\eta}_2(\tau)\bar{F}_1(\tau)
$$

and a stationary point of this optimization satisfies

$$
\begin{aligned}
0 &= \bar{F}_1(\tau) + \bar{\eta}_2'(\tau)\bar{F}_1(\tau) - \bar{\eta}_2(\tau)f_1(\tau) \\
&= \bar{F}_1(\tau)\bar{\eta}_2(\tau)\left[\frac{1 + \bar{\eta}_2'(\tau)}{\bar{\eta}_2(\tau)} - \frac{f_1(\tau)}{\bar{F}_1(\tau)}\right]\,.
\end{aligned}
$$

Thus, under Assumption 4, the optimal stopping time condition is

$$
\frac{1 + \bar{\eta}_2'(\tau)}{\bar{\eta}_2(\tau)} = \frac{f_1(\tau)}{\bar{F}_1(\tau)}\,.
$$

Proof of Theorem 4

We put together the stochastic discrete time argument with the deterministic continuous time argument developed above to give a formal proof. Towards this purpose, we must formulate the optimal stopping problem as a free boundary problem and then solve it for the optimal rule. The theory of free boundary problems and optimal stopping is given in detail in [38].

A value function must satisfy the following free boundary problem

$$
\bar{\eta}_2(t) \geq V(t)\,, \tag{38a}
$$

$$
0 = 1 + V'(t) - \frac{f_1(t)}{\bar{F}_1(t)}V(t) \quad \text{on} \quad \{V(t) < \bar{\eta}_2(t)\}\,. \tag{38b}
$$

It is show in Section 2.2 of [38] that a solution of (38) defines the value of a policy that stops on the set $S = \inf\{t : V(t) \geq \bar{\eta}_2(t)\}$. Assuming that $V(t) < \bar{\eta}_2(t)$, the o.d.e. (38b) can be solved as follows

$$
\begin{aligned}
&0 = 1 + V'(t) - \frac{f_1(t)}{\bar{F}_1(t)}V(t) \\
&\implies f_1(t)V(t) - \bar{F}_1(t)V'(t) = \bar{F}_1(t) \\
&\implies \frac{d}{dt}(-\bar{F}_1(t)V(t)) = \bar{F}_1(t)
\end{aligned}
$$

$$\implies \bar{F}_1(t)V(t) = -\int_0^t \bar{F}_1(s)ds + A$$
$$\implies V(t) = \frac{A}{\bar{F}_1(t)} - \frac{1}{\bar{F}_1(t)}\int_0^t \bar{F}_1(s)ds$$

where A is a constant which we will specify shortly.

We now investigate times $\tau \in [0, \infty]$ where $V(\tau) = \bar{\eta}_2(\tau)$. Substituting the above expression for $V(t)$, we notice that

$$\bar{\eta}_2(\tau) = \frac{A}{\bar{F}_1(\tau)} - \frac{1}{\bar{F}_1(\tau)}\int_0^\tau \bar{F}_1(s)ds \implies A = \bar{\eta}_2(\tau)\bar{F}_1(\tau) + \int_0^\tau \bar{F}_1(s)ds\,.$$

Thus, we see that the solutions to the free boundary problem, for which there exists a time with $V(\tau) = \bar{\eta}_2(\tau)$, take the form

$$\begin{aligned} V_\tau(t) &= \bar{\eta}_2(\tau)\frac{\bar{F}_1(\tau)}{\bar{F}_1(t)} + \int_t^\tau \bar{F}_1(s)ds \\ &= \bar{\eta}_2(\tau) + \frac{1}{\bar{F}_1(t)}\int_t^\tau \bar{\eta}_2(\tau)f_1(s) + \bar{F}_1(s)ds\,, \quad t \le \tau\,. \end{aligned} \tag{39}$$

Here we write $V_\tau(t)$ to make the dependence on τ explicit. We now require the minimal solution, setting $t = 0$ and differentiating with respect to τ gives

$$\begin{aligned} \partial_\tau V_\tau(0) &= \bar{\eta}_2'(\tau)\bar{F}_1(\tau) - \bar{\eta}_2(\tau)f_1(\tau) + \bar{F}_1(\tau) \\ &= \bar{\eta}_2(\tau)\bar{F}_1(\tau)\left[\frac{1+\bar{\eta}_2'(\tau)}{\bar{\eta}_2(\tau)} - \frac{f_1(\tau)}{\bar{F}_1(\tau)}\right]. \end{aligned}$$

The term $\bar{\eta}_2(t)\bar{F}_1(\tau)$ is positive while the term in square brackets is monotone increasing from Assumption 4. Thus, from the term in square brackets above, we see that the condition is that τ^* is the minimal value such that the term in square brackets is positive. That is

$$\tau^* = \min\left\{\tau \ge 0 : \frac{1+\bar{\eta}_2'(\tau)}{\bar{\eta}_2(\tau)} \ge \frac{f_1(\tau)}{\bar{F}_1(\tau)}\right\}.$$

Thus (39) with $\tau = \tau^*$ characterizes the solution up until time τ^*. We note that for time $t > \tau^*$ the solution must satisfy $V(t) = \bar{\eta}_2(t)$. This can be seen because of the following argument. If $V(t) = \bar{\eta}_2(t)$ did not hold for all $t \ge \tau^*$ then we can choose a time t such that $V(t) = \bar{\eta}_2(t)$, but the value function is strictly smaller immediately after time t; however, then under condition (38b)

$$V'(t) = 1 - \frac{f_1(t)}{\bar{F}_1(t)}V(t) = 1 - \frac{f_1(t)}{\bar{F}_1(t)}\bar{\eta}_2(t) > \bar{\eta}_2'(t)$$

where the inequality above holds since

$$\frac{1+\bar{\eta}_2'(t)}{\bar{\eta}_2(t)} > \frac{f_1(t)}{\bar{F}_1(t)}$$

for $t > \tau^*$. Thus we have $V'(t) > \bar{\eta}_2'(t)$. So we see that this leads to a contradiction since we assumed the function $V(t)$ decreases below $\bar{\eta}_2(t)$ immediately after time t. This proves that $V(t) = \bar{\eta}_2(t)$ for all $t \ge \tau^*$.

From this we see that the minimal value function solving the free boundary problem is

$$V(t) = \begin{cases} \bar{\eta}_2(\tau^*) + \frac{1}{\bar{F}_1(t)}\int_t^{\tau^*}[\bar{\eta}_2(\tau^*)f_1(s) + \bar{F}_1(s)]ds & t \le \tau^* \\ \bar{\eta}_2(t) & t > \tau^* \end{cases}$$

where τ^* is given above, which implies that the optimal stopping set is $S := \{t : V(t) = \bar{\eta}_2(t)\} = \{t : t \ge \tau^*\}$. Thus, it is optimal to stop at time τ^* as required.

Finally, we note that if η_2 is independent of η_1 then $\bar{\eta}_2(t)$ is a constant, $\mathbb{E}[\eta_2]$, and the optimal stopping condition reduces to the condition (20).