Response Time Distribution in a Tandem Pair of Queues with Batch Processing

P. G. HARRISON and J. BOR, Imperial College London

Response time density is obtained in a tandem pair of Markovian queues with both batch arrivals and batch departures. The method uses conditional forward and reversed node sojourn times and derives the Laplace transform of the response time probability density function in the case that batch sizes are finite. The result is derived by a generating function method that takes into account that the path is not overtake-free in the sense that the tagged task being tracked is affected by later arrivals at the second queue. A novel aspect of the method is that a vector of generating functions is solved for, rather than a single scalar-valued function, which requires investigation of the singularities of a certain matrix. A recurrence formula is derived to obtain arbitrary moments of response time by differentiation of the Laplace transform at the origin, and these can be computed rapidly by iteration. Numerical results for the first four moments of response time are displayed for some sample networks that have product-form solutions for their equilibrium queue length probabilities, along with the densities themselves by numerical inversion of the Laplace transform. Corresponding approximations are also obtained for (non-product-form) pairs of "raw" batch-queues—with no special arrivals—and validated against regenerative simulation, which indicates good accuracy. The methods are appropriate for modeling bursty internet and cloud traffic and a possible role in energy-saving is considered.

CCS Concepts: • **Networks** → *Network performance modeling*; *Network performance analysis*; • **Mathematics of computing** → *Markov networks*; **Queueing theory**; **Markov processes**;

Additional Key Words and Phrases: Response time distributions, queuing networks with batches, generating function method, moments

ACM Reference format:
P. G. Harrison and J. Bor. 2021. Response Time Distribution in a Tandem Pair of Queues with Batch Processing. *J. ACM* 68, 4, Article 22 (May 2021), 41 pages.
https://doi.org/10.1145/3448973

1 INTRODUCTION

Queuing systems with batches are appropriate for modeling *burstiness* in physical networks, which has been widely observed in internet and storage network traffic for some years, for example, in IP and data transfer networks [27, 28]. Tandem queues have been used to model wireless sensor networks to find optimal centralized or decentralized sensor scheduling policies [29], and the ability to model batches increases their domain of application to far more realistic systems. Such models may also be relevant in power management, where energy can be saved by using devices

Authors' addresses: P. G. Harrison, Department of Computing, Imperial College London, London SW7 2AZ; email: pgh@ic.ac.uk; J. Bor, Department of Computing, Imperial College London, London SW7 2AZ; email: julianna.bor16@imperial.ac.uk.

0004-5411/2021/05-ART22 $15.00
https://doi.org/10.1145/3448973

with multiple power levels of operation. Increasing the burstiness of traffic lengthens idle periods during which devices may be powered down.

Stochastic performance modeling of queues and networks of queues with batches is therefore important. Such networks, which we call *batch-networks*, do not have product-forms for their equilibrium queue length distributions, even when they are Markovian, i.e., have Poisson arrivals of batches of tasks and exponential task-service times. However, it has been shown that they do have product-forms when there are additional Poisson batch arrival streams at empty queues as well as partial batch departures that leave behind an empty queue [2, 19]. The product-form is simply computed using results in [14]. One good mathematical account of the general field of stochastic modeling that goes to the appropriate depth is [34].

The present article has two main take-aways;

(1) First, the *response time distribution is obtained exactly* (up to Laplace-Stieltjes transform) for a two-node, product-form tandem network with batches, when there are certain additional traffic streams at empty queues. This is a mathematically complex problem with two novel aspects in its solution, discussed below. Its importance lies in the fact that it is the first such result for batch-networks of more than one node (to the authors' best knowledge), thus providing a benchmark for approximate models and simulation.
(2) Second, using the recent result of [17], which provides an approximate semi-product-form for the joint queue length probabilities at equilibrium, the response time distribution is obtained for *unmodified tandem batch-networks of two queues* to arbitrary accuracy (within practical computational limits). This is also a new result to our knowledge, and one with good potential for practical application since it requires no additional traffic streams. Possible applications are discussed in more detail in Section 7 but include networks composed of high-speed switches serving randomly arriving large messages that are broken up into packets of various sizes. Such networks are widespread in communication systems, storage networks, and internet traffic. However, the focus of the article is theoretical rather than specific case studies.

Both of these contributions provide significant advances in the study of response time distributions, which has been an important research area for decades since the seminal papers of the 1980s [5–7, 16, 22, 30, 38, 40]. Among other results on non-product-form tandem networks is Kella and Whitt's fluid model of [23], which analyzes a linear sequence of "pipes" linking buffers, the content of which is described as a volume of fluid. Results for the distribution of content is obtained at equilibrium, and the model is solved explicitly for a tandem network of two nodes with no product-form. Structural properties such as stability and tightness of their content process are considered in [24]. In a queuing context, the pipes correspond to servers that are assumed to work at a constant rate. This is a strong restriction, but it is compensated for by the generality of the arrival process, which can be a Lévy process, only requiring independent increments. Relevant to our work, the latter article considers a more specific input process that allows a fraction of the output from each buffer to be discarded rather than passed on to another buffer. However, neither article considers response times, apart from their mean value via Little's result [16, 34].

In Section 2, we consider a batch-queue with special arrivals that endow the queue with a geometric equilibrium queue length probability distribution. We derive the forward and reversed task sojourn time densities, given the numbers of tasks in front of and behind a "tagged" task that is tracked until its departure. This result forms the basis of the response time analysis presented for a two-node tandem batch-network in Section 3, where response time is considered as the sum of the reversed sojourn time in queue 1 and the forward sojourn time in node 2, conditioned on the

state of the system at the instant the tagged task transits between the nodes; this is the approach of the RCAT methodology [13] and was first used in [1].

In batch-networks, tasks arriving at a node after the tagged task can influence the latter's progress so that the network is not overtake-free and so not separable [40]. Our solution solves for a vector of generating functions and requires the singularities of a certain matrix to be found. This is significantly more complex than traditional problems of similar type, where only the zeros of a scalar-valued expression are needed [6, 8, 22, 30]. Although a vector of unknown *constants* is required in [32], this is easily obtained as a routine solution of independent linear equations.

The (joint) Laplace transform of the joint density of the sojourn times at the two queues can be expressed as the geometrically weighted sum of the pairwise product of the i^{th} coefficients in two generating functions corresponding to each node; these coefficients are not independent and so the sum does not separate. However, the problem is transformed into an integral around a circle centered on the origin in the complex plane of the product of the generating functions themselves, giving a closed-form solution. This solution is evaluated numerically.

A recurrence formula is derived in Section 4 to obtain arbitrary moments of response time by differentiation of the Laplace transform at the origin, leading to a set of problems of the above type, one for each moment required. The moments can then be computed rapidly by a simple iteration. Numerical results for the mean, standard deviation, skewness, and kurtosis of response times are given in Section 5, along with the densities themselves, computed by numerical inversion of their Laplace transform. In Section 6 we use the same method to find the response time distribution in an *unmodified* batch-network, obtaining an approximate solution that compares very favorably with the results of a regenerative simulation. Notice that the special batch arrivals are used *solely* to provide a product-form solution for the queue length probabilities in the exact analysis; they are absent in the second model. Possible applications are considered in Section 7 and the article concludes in Section 8.

A "roadmap" of the article is provided in Figure 1 as a preview of the article's structure, highlighting novelty and distinguishing theoretical content from implementation of results. A glossary of terms used in the article is provided in Table 1 and a roadmap showing the dependencies among the theorems and propositions used in the article is provided in Appendix H.

2 SOJOURN TIME IN A GEOMETRIC BATCH-QUEUE

Our model of batch transitions in a single-server, Markovian queue is as follows (refer to Figure 2):

- The state space $\mathcal{S}$ of the queue is the set of non-negative integers, representing the possible numbers of tasks in the queue.
- *Normal* batch arrivals are Poisson with size $k \geq 1$ and are represented by state transitions with constant rate $a_k : i \to i + k \quad (i \geq 0)$, i.e., from states i to $i + k$.
- Additional *special* Poisson batch arrivals of size $k \geq 1$ to an empty queue only are represented by state transitions with rate $a_{0k} : 0 \to k$.
- *Full* batch departures of size $k \geq 1$ (due to completions of exponential service times) are represented by transitions with constant rate $d_k : i + k \to i \quad (i \geq 0)$.
- *Partial* batch departures of size $k \geq 1$, leading to an empty queue, are represented by transitions with rate $d_{k0} : k \to 0$, where $d_{k0} = \sum_{j=1}^{\infty} d_{k+j}$. The rate d_k can be regarded as the departure rate for *intended batch size* k: if the queue length is k or greater, the actual size of the batch leaving the node is k, but if not, the actual batch size is equal to the queue length.
- Arrival and departure batch sizes are independent.
- The task-ordering in the queue is **first come first served (FCFS)**.

Thus, in customized Kendall notation, the batch-queue we consider is $M^{B,B_0}/M^{B,B_0}/1$, where the superscript B denotes batch arrivals in any state of the queue and (independent) batch

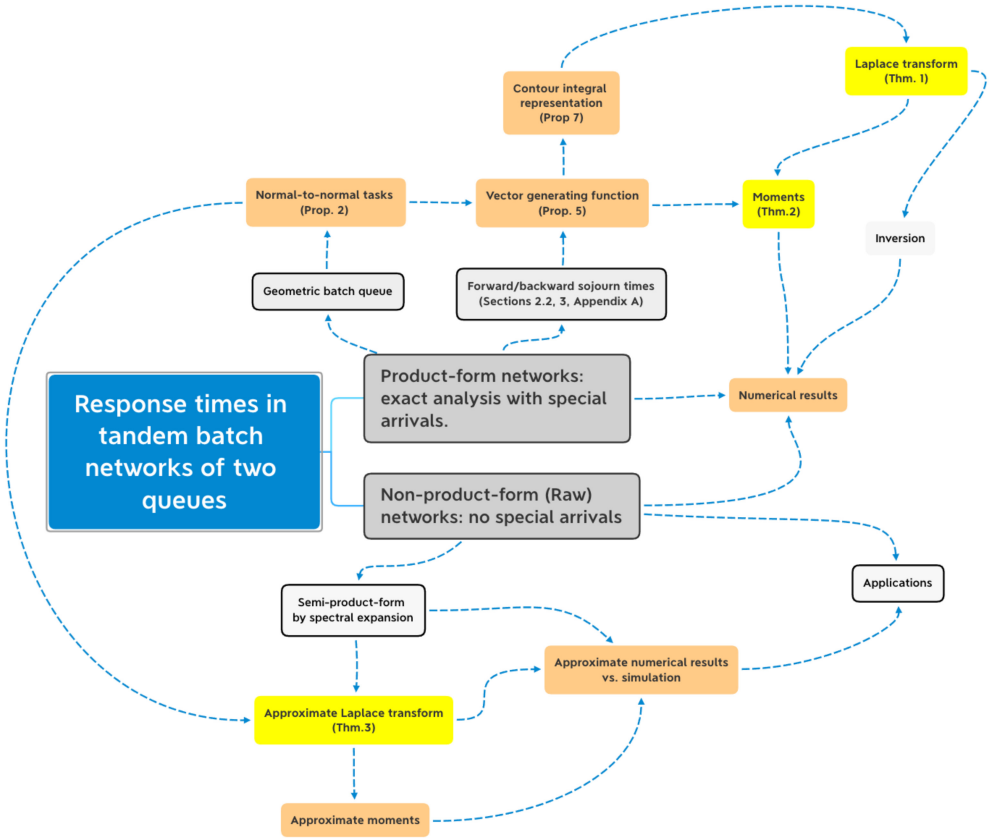

Fig. 1. Roadmap of article.

departures from any non-empty state, and B_0 denotes independent batch arrivals to an empty queue and independent batch departures that result in an empty queue. This queue models the effects of *buffering*: tasks are first grouped together into batches prior to input to the queue; then the server groups them into new batches for processing and output. In the above definition, the time to process an output buffer of size k is exponential with mean d_k^{-1}. Many network components operate in this way, where buffering at input ports and output ports is asynchronous, for example, routers supporting MPEG traffic or SSD controllers [11, 36]. An alternative to (re)buffering has fixed batches, each containing the same tasks throughout their progress in a network. Then, the basic unit of work can be redefined as the service time pertaining to all the tasks in a batch, in terms of the given distributions of the task-service time and the batch size. This can greatly simplify the model; for example, an $M^B/M^B/1$ queue can be modeled as $M/G/1$.

In a batch-queue considered in isolation, what happens to completing partial batches, with smaller-than-intended size, does not matter, as long as they depart and leave a queue of length zero behind. However, as we shall see in Sections 2.2, 6, and 7, their subsequent passage in a larger network can be specified in various ways, giving very different network semantics and response time characteristics.

Rate-generating functions are defined for each batch transition as follows:

$$A(z) = \sum_{k=1}^{\infty} a_k z^k \quad A_0(z) = \sum_{k=1}^{\infty} a_{0k} z^k \quad D(z) = \sum_{k=1}^{\infty} d_k z^k.$$

Table 1. Glossary of Terms and Definitions

Term	Description
Sojourn time	The time taken for an individual task to pass through a queuing node or the whole network.
Response time	Network sojourn time.
GBQ	Geometric batch-queue.
RCAT	Reversed Compound Agent Theorem of [13].
LST	Laplace-Stieltjes transform.
Normal batch	A batch that arrives in either queue as part of the offered workload.
Normal task	A task anywhere in the network that originally arrived as part of a normal batch.
Special batch	An additional batch specifically introduced to obtain a product-form solution.
Special task	A task that originally arrived as part of a special batch. Special batches and tasks are not of interest per se but they do influence the performance of the normal tasks.
Model parameters and generating functions	
a_j, a_{0j}, d_j	Arrival, special arrival, and service rates for batch size j.
$\tilde{a}_j(n)$, $\tilde{d}_j(n)$	Arrival and service rates in the reversed process at queue length n for batch size j.
$x_{1,i}$, $c_{1,i}$, $\vec{e}_{1,i}$, $x_{2,i}$, $c_{2,i}$, $\vec{e}_{2,i}$	Parameters from the semi-product-form, SEM model of [17].
$A_i(z)$, $D_i(z)$, $\tilde{A}_i(z)$, $\tilde{D}_i(z)$	Generating functions for the arrival and service rates at node i (forward and reversed).
$H_j(x;\theta)$, $\tilde{H}_j(x;\theta)$	Generating function for the forward and reversed sojourn times at the first node, given j tasks behind the tagged task.
$G_j(x, z;\theta)$	Generating function for the forward sojourn time at the second node, given j tasks behind the tagged task.
T^*	Generating function of two-node response time (Theorem 1).
Random variables and related functions	
$X(t)$	Cumulative probability distribution function (cdf) of the generic random variable X.
$X^*(\theta)$	LST of $X(t)$, i.e., $\mathbb{E}[e^{-\theta X}] = \int_0^\infty e^{-\theta t} dX(t)$.
$T_F(\theta)$	Sojourn time random variable in a single queue for the first task in a batch.
$T_R(\theta)$	Sojourn time random variable in a single queue for a random task in a batch.
R_m	Remaining sojourn time when there are m tasks ahead of the tagged task.
T	Sojourn time at a single node of a normal-to-normal task in a random batch position.
S_1	Task sojourn time in queue 1.
S_2	Task sojourn time in queue 2.
S	Abbreviation of S_2 in Section 3.1.
J, K	Numbers of tasks behind and in front of the tagged task.
I	Number of tasks in queue 1.
L	Number of tasks ahead of the tagged batch in queue 2.
M	Number of tasks behind the tagged task in its batch.
N	Number of tasks ahead of the tagged task in its batch.
$\tilde{\bullet}$	Denotes the reversed process.
Matrices and vectors	
$\mathbb{L}$, $\mathbb{Y}$, $\mathbb{D}$, $\mathbb{M}$, $\mathbb{K}$, $\vec{v}$	Terms used in the formula for $\vec{H}(x;\theta)$.
$\mathbb{J}$, $\mathbb{M}_1$, $\mathbb{M}_2$, $\mathbb{K}_1$, $\mathbb{K}_2$, $\vec{E}$	Terms used in the formula for $\vec{G}(x, z;\theta)$.

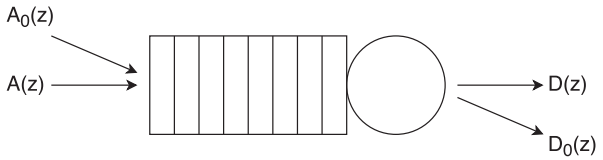

Fig. 2. The batch-queue model.

We assume that $A(1), A_0(1), D(1) < \infty$, to avoid null mean state holding times (i.e., infinite total instantaneous transition rate out of a state). The functions $A(z), A_0(z), D(z)$ are therefore absolutely convergent and analytic inside the unit disk, which lies inside their circles of convergence. Notice that the superposed normal batch arrival process (i.e., batches of any size) is Poisson with rate $A(1)$ and the effective service time of a batch (i.e., time to the next batch service completion of any size) is exponential with parameter $D(1)$ in all states.

The following proposition gives conditions for the length of the queue (number of tasks) to have a geometric equilibrium probability distribution, along with its parameter in that case. The queue thus defined is then called a *geometric batch-queue*, abbreviated to GBQ. As a consequence, networks of such queues, with appropriately defined special arrival processes at each queue, have product-forms by application of the **Reversed Compound Agent Theorem (RCAT)** [13, 14].

Proposition 1. *A batch-queue with $A(1), D(1) < \infty$ has geometrically distributed equilibrium queue length probabilities with parameter $\rho < 1$, $\pi_n = (1-\rho)\rho^n$ for $n \geq 0$, iff*

$$A_0(z) = \frac{[A(1) + D(1) - D(\rho)]\rho z - A(z)}{1 - \rho z} \tag{1}$$

for $|z| < \min(\rho^{-1}, R)$, where R is the radius of convergence of the series $A(z)$.

If $A(\rho^{-1}) < \infty$, then ρ is the unique real solution of the equation

$$A(\rho^{-1}) + D(\rho) = A(1) + D(1)$$

in the interval $(0, 1)$, whereupon we may write

$$A_0(z) = \frac{A(\rho^{-1})\rho z - A(z)}{1 - \rho z}. \tag{2}$$

Proof. The proposition was first proved in part in [18] and then more fully in [14]. □

We consider the case of *finite batch sizes*, where the generating functions $A(z)$ and $D(z)$ are finite sums. We then define n_a and n_d to be the maximum batch sizes for arrivals and departures, respectively. Then $A(\rho^{-1}) < \infty$ and the *rate equation* $A(\rho^{-1}) + D(\rho) = A(1) + D(1)$ uniquely determines the parameter ρ. In this case, the special arrivals into the empty queue have rates given by $A_0(z) = [A(\rho^{-1})\rho z - A(z)]/(1 - \rho z)$.

The batch-queue so defined is called a *minimal discard queue*. It is a discard queue by virtue of the above definition of partial batches via d_{k0}, which are "discarded" into a special output stream when there are fewer than k tasks in the queue, and "minimal" in that the degree of the polynomial $A_0(z)$ is minimized, since the denominator in the definition of $A_0(z)$ cancels—giving degree one less than the degree of $A(z)$; see [14].

2.1 Forward and Reversed Task Sojourn Times

The forward and reversed sojourn times in a batch-queue are important since, when considering response times in a tandem pair of batch-queues in the next section, we use the reversed sojourn time at the first node and the forward sojourn time at the second node.

The first task in a normal arriving batch will see n tasks in the queue with probability $(1-\rho)\rho^n$, by the **random observer property (ROP)**,[1] and the LST of the sojourn time distribution of the first task in a batch is shown to be $T_F^*(\theta) = \frac{D(1)-D(\rho)}{\theta+D(1)-D(\rho)}$ in [18]. Surprisingly, therefore, the first task in a batch has an exponentially distributed sojourn time with parameter $D(1) - D(\rho)$.

A *random task* is a task whose position within its batch is uniformly distributed over the range $[1, B]$, where B is the size of the batch. A random task therefore joins the queue in position $K + 1$, where $K = N + L$, L is the number of tasks in the queue just before the batch's arrival instant, and N is the number of tasks ahead of the random task in its arriving batch. Thus, N is a discrete forward (or backward) recurrence time so that $\mathbb{P}(N = n) = \sum_{i=n+1}^{\infty} a_i/\dot{A}(1)$, where $\dot{A}(1)$ denotes the derivative of the generating function $A(z)$ at $z = 1$, which is proportional to the mean batch size of normal arrivals. Now, $\mathbb{P}(L = \ell) = (1-\rho)\rho^\ell$ by the ROP and so the LST of the sojourn time distribution of a random task in a batch is (say)

$$T_R^*(\theta) = (1-\rho)\sum_{k=0}^{\infty}\sum_{n=0}^{k} b_n \rho^{k-n} R_k^*(\theta),$$

where $b_n = \sum_{i=n+1}^{\infty} a_i/\dot{A}(1)$ and the random variable R_k denotes the remaining sojourn time of a task in queue position $k + 1$, with probability distribution function $R_k(t)$ having LST $R_k^*(\theta) = \mathbb{E}[e^{-\theta R_k}]$, shown to be the coefficient of x^k in $\frac{D(1)-D(x)}{(1-x)(\theta+D(1)-D(x))}$ in Proposition 6 of [18].[2]

At equilibrium, the reversed process of a discard batch-queue has the same geometric queue length probabilities as the forward queue. Therefore, the reversed process is also a geometric batch-queue with parameter ρ and rate-generating functions $\tilde{A}(z) = D(\rho z), \tilde{D}(z) = A(\rho^{-1}z)$; see Proposition 2 of [18]. It is a discard queue since $\tilde{D}_0(z) = A_0(\rho^{-1}z) = (zA(\rho^{-1}) - A(\rho^{-1}z))/(1-z) = (z\tilde{D}(1) - \tilde{D}(z))/(1-z)$ by Equation (2), so that, comparing the coefficients of z^k, $\tilde{d}_{k0} = \sum_{i=1}^{\infty}\tilde{d}_i - \sum_{i=1}^{k}\tilde{d}_i = \sum_{i=k+1}^{\infty}\tilde{d}_i$. It is also minimal because it is evident that $\tilde{A}(1) + \tilde{D}(1) = \tilde{A}(\rho^{-1}) + \tilde{D}(\rho)$.

2.2 Normal-to-Normal Tasks

To investigate sojourn times in a stochastic model, a particular task is tracked as it progresses from its arrival instant in some state of the model to its departure instant from another state. This task is often called the *tagged task*. In a network of discard batch-queues, when a node generates a service completion with intended batch size exceeding the current queue length, all the tasks in the queue at that instant leave the network as a partial batch. As a result, batches arriving in the queue after the tagged task may prevent it from departing the network in a partial batch by virtue of the fact that on the tagged task's completion of service, there aren't enough other tasks in the queue to make up the full intended batch. These later batches therefore influence the tagged task's progress and so the network is not "overtake-free" in the sense of [40]. This is not to say that literal re-sequencing of tasks can take place, only that later-arriving tasks can have an effect on earlier ones. Overtake-free networks of M/M/1 queues have special properties, such as product-forms for their response time distributions [12, 26], whereas networks that are not overtake-free are much harder to solve [6, 22, 30]. The present model falls in the latter category.

Discard batch-queues were used in an "assembly-transfer network" model in Chapter 8 of [2], which is appropriate for models of certain manufacturing systems. The interpretation is that intended batches of size less than or equal to the queue length are full batches, but the partial batches are incomplete—with a size equal to the queue length, which is less than the intended

[1] See, for example, [16, 34] for ROP and the related result PASTA.

[2] As a matter of notation in general, univariate distribution functions take the same name as their random variable and use asterisks to denote their LSTs throughout this article; see the glossary of terms in Table 1.

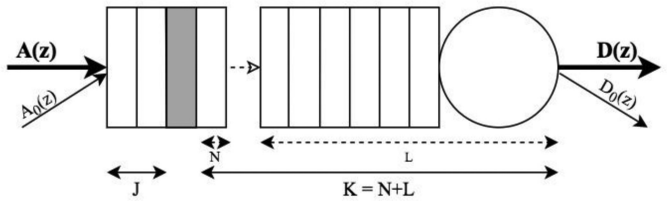

Fig. 3. Batch-queue at the arrival instant of the tagged task, with J tasks behind in the arriving batch, 1 (in this example) in front in the arriving batch, and K in total ahead in the queue joined, i.e., $K-1$ already in that queue.

batch size k. These are regarded as corrupted and are discarded in the assembly line, departing at rates d_{0k} $(1 \le k \le n_d - 1)$. Many printers operate in a similar way after a "jam" has been released, discarding the sheets already printed in the job being processed. Other modes of operation are more common in practice, e.g., forwarding the partial batch to the same node as the intended batches, and these modes can be modeled by the approximate method described in Section 6. However, first we choose the discard mode since that is the only one that has a product-form for the equilibrium queue length probabilities, facilitating the exact analytical solution for response time distribution we derive in Section 3.1.

Response time (or network sojourn time) is the time spent by a normal task in the network, i.e., the time elapsed between the arrival of its containing batch and its departure in a full batch from the last queue in the network; it relates to a sequence of *normal* batch services. The sojourn times considered so far relate to the *unconditional* time spent by any task, regardless of whether it departs in a normal or a special batch, so we now need to consider the joint probability that a normally arriving task has sojourn time $T \le t$ for some value t *and* departs in a normal batch—this defines a *normal-to-normal* task.

Referring to Figure 3, the random variables J and K denote the numbers of tasks *behind* and *in front of* the tagged task, respectively. We seek the probabilities $\gamma_{kj}(t) = P(T < t, \Im = 1 | K = k, J = j)$, where $\Im$ is the indicator function of the event that the departing batch is normal and T is the sojourn time random variable. Let $H_j(x;\theta) = \sum_{k=0}^{\infty} \gamma^*_{kj}(\theta) x^k$ be the generating function of the LST of this conditional probability distribution, for each j.

Proposition 2. *For $j \ge 0$, $H_j(x;\theta)$ satisfies the recurrence relation*

$$[A(1) + D(1) - D(x) + \theta] H_j(x;\theta) = \sum_{s=1}^{n_a} a_s H_{j+s}(x;\theta) + \sum_{i=0}^{n_d-1} \sum_{s=i+1}^{i+j+1 \wedge n_d} d_s x^i, \tag{3}$$

where $\wedge$ is the infix operator for the minimum of two numbers.[3]

Proof. Given a queue length of $k + j + 1$ at time t, with j tasks behind and k tasks in front of the tagged task, consider the infinitesimal interval $(t, t + h]$. The following events can occur:

(1) normal arrival;
(2) normal departure without the tagged task;
(3) normal departure containing the tagged task.

Note that there cannot be any special arrivals because the queue contains the tagged task so is non-empty. Since the arrival process is Poisson and service times are exponential random variables, the

[3] $\wedge$ has the lowest operator precedence of the arithmetic operators, so that, for example, $i + j + 1 \wedge n_d \equiv (i + j + 1) \wedge n_d$.

probability that a batch arrives in an interval of length h is $\lambda h + o(h)$, and similarly for batch service completions. Thus, we have, for $k, j \geq 0$:

$$\gamma_{kj}(t+h) = h\sum_{s=1}^{n_a} a_s \gamma_{k,j+s}(t) + h\sum_{s=1}^{k\wedge n_d} d_s \gamma_{k-s,j}(t) + h\sum_{s=k+1}^{k+j+1\wedge n_d} d_s + [1 - hA(1) - hD(1)]\gamma_{kj}(t) + o(h),$$

where the third term on the right-hand side describes the tagged task leaving the node in a full batch within time t. Batches larger than the queue length $k + j + 1$ lead to $\Im = 0$, i.e., failure to leave in a normal batch. After rearranging, dividing by h, and taking the limit $h \to 0$, we take the LST of both sides to obtain

$$[\theta + A(1) + D(1)]\gamma^*_{kj}(\theta) = \sum_{s=1}^{n_a} a_s \gamma^*_{k,j+s}(\theta) + \sum_{s=1}^{k\wedge n_d} d_s \gamma^*_{k-s,j}(\theta) + \sum_{s=k+1}^{k+j+1\wedge n_d} d_s,$$

noting that the LST of the derivative of a **cumulative probability distribution function (cdf)** is the parameter θ multiplied by the LST of that cdf. Multiplying by x^k and then summing from $k = 0$ to ∞, we get

$$[\theta + A(1) + D(1)]H_j(x;\theta) = \sum_{s=1}^{n_a} a_s H_{j+s}(x;\theta) + \sum_{k=0}^{\infty}\sum_{s=1}^{k\wedge n_d} d_s \gamma^*_{k-s,j}(\theta)x^k + \sum_{k=0}^{\infty}\sum_{s=k+1}^{k+j+1\wedge n_d} d_s x^k,$$

where the second term on the right has a closed form:

$$\sum_{k=0}^{\infty}\sum_{s=1}^{k\wedge n_d} d_s \gamma^*_{k-s,j}(\theta)x^k = \sum_{s=1}^{n_d}\sum_{k=s}^{\infty} d_s \gamma^*_{k-s,j}(\theta)x^k = \sum_{s=1}^{n_d}\sum_{k=0}^{\infty} d_s \gamma^*_{k,j}(\theta)x^{k+s} = D(x)H_j(x;\theta).$$

Hence, we get the stated equation. □

Notice that as soon as there are at least $n_d - 1$ tasks behind the tagged one, a new arrival can no longer influence the tagged task's sojourn time, which means $H_j(x;\theta) = H_{n_d-1}(x;\theta)$ for all $j \geq n_d - 1$.

The above proposition can be written in matrix form as

$$\mathbb{L}(x;\theta)\vec{H}(x;\theta) = \mathbb{Y}\mathbb{D}\vec{v}(x), \tag{4}$$

where $\vec{v}(x) = (1, x, \ldots, x^{n_d-1})$, $\mathbb{L}(x;\theta) = (A(1) + D(1) - D(x) + \theta)\mathbb{I} - \mathbb{M} - \mathbb{K}$, $\mathbb{I}$ is the identity $n_d \times n_d$ matrix, and $\mathbb{D}, \mathbb{Y}, \mathbb{M}, \mathbb{K}$ are $n_d \times n_d$ matrices defined as follows:

$$\mathbb{D} = \begin{bmatrix} d_1 & \ldots & d_{n_d-1} & d_{n_d} \\ d_2 & \ldots & d_{n_d} & 0 \\ \vdots & \iddots & \iddots & \vdots \\ d_{n_d} & 0 & \ldots & 0 \end{bmatrix} \qquad \mathbb{Y} = \begin{bmatrix} 1 & 0 & \ldots & & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ & \vdots & & \vdots & \\ 1 & 1 & \ldots & & 1 \end{bmatrix}$$

$$\mathbb{M}: \begin{cases} m_{ij} = a_{j-i} & 0 \leq i \leq n_d - 1, i+1 \leq j \leq n_a + i \wedge n_d - 1 \\ 0 & \text{otherwise} \end{cases}$$

(*upper triangular*)

$$\mathbb{K}: \begin{cases} k_{ij} = \sum_{s=n_d}^{n_a+i} a_{s-i} & 0 \leq i \leq n_d - 1, j = n_d - 1 \\ 0 & \text{otherwise} \end{cases}$$

(*only the last column is non-zero*).

The matrix $\mathbb{L}(x;\theta)$ is upper triangular with positive diagonal elements when $0 \le x < 1$ since then $D(1) > D(x)$. It is therefore invertible when $0 \le x < 1$, and also when $x = 1$ and $\theta > 0$; when $x = 1$ and $\theta = 0$, the last diagonal element is zero. Other than in this one case, which does not arise in practice, we can therefore solve the equation for $\vec{H}(x;\theta)$ to find

$$\vec{H}(x;\theta) = \mathbb{L}(x;\theta)^{-1}\mathbb{Y}\mathbb{D}\vec{v}(x).$$

Now let $\pi_{kj} = \mathbb{P}(K = k, J = j)$ be the probability that there are k tasks in front of and j tasks behind the tagged task *at its arrival instant*. Then, $\pi_{kj} = (1-\rho)\sum_{s=0}^{k}\phi_{js}\rho^{k-s}$ by the random observer property, where ϕ_{js} is the probability that there are s tasks in front of and j tasks behind the tagged one *in its arrival batch*. Hence, for the first task in the batch, $\phi_{js} = \delta_{s0}a_{j+1}/A(1)$, and for a random task in the batch, $\phi_{js} = a_{s+j+1}/\dot{A}(1)$; we consider the latter case.[4]

Proposition 3. *The LST of the probability distribution function of the sojourn time T of a normal-to-normal task in a random batch position is*

$$T^*(\theta) = (1-\rho)\sum_{j=0}^{n_a-1}\sum_{s=0}^{n_a-j-1}\rho^{-s}\phi_{js}\left[H_{j\wedge n_d-1}(\rho;\theta) - \sum_{k=0}^{s-1}\frac{\rho^k}{k!}\left.\frac{\partial^k H_{j\wedge n_d-1}(x;\theta)}{\partial x^k}\right|_{x=0}\right]. \tag{5}$$

Proof. Noting that $\phi_{js} = 0$ whenever $s + j \ge n_a$,

$$\begin{aligned}
T^*(\theta) &= \sum_{j=0}^{n_a-1}\sum_{k=0}^{\infty}\pi_{kj}\gamma^*_{kj}(\theta)\\
&= (1-\rho)\sum_{j=0}^{n_a-1}\sum_{s=0}^{n_a-j-1}\sum_{k=s}^{\infty}\phi_{js}\rho^{k-s}\gamma^*_{kj}(\theta)\\
&= (1-\rho)\sum_{j=0}^{n_a-1}\sum_{s=0}^{n_a-j-1}\phi_{js}\rho^{-s}\left[\sum_{k=0}^{\infty}\rho^k\gamma^*_{kj}(\theta) - \sum_{k=0}^{s-1}\rho^k\gamma^*_{kj}(\theta)\right].
\end{aligned}$$

The result now follows from the definition of $H_j(\rho;\theta)$, noting that $\frac{1}{k!}\frac{\partial^k H_{j\wedge n_d-1}(x;\theta)}{\partial x^k}$ at $x = 0$ is the coefficient of x^k, i.e. $\gamma^*_{kj}(\theta)$. □

3 RESPONSE TIMES IN TANDEM BATCH-NETWORKS

So far we have obtained results for one batch-queue on its own, augmented with special arrivals and departures that may be incomplete. In this section we consider a tandem pair of such queues, as per Figure 4. Each queue has external arrivals, both normal and special, and each has incomplete departures, which exit the network, leaving their queue empty. Normal, or intended, departures are passed from node 1 to node 2 and leave node 2 as completed tasks that finish their sojourn in the network; these are the only ones that would contribute a response time in a simulation, for example. This specification gives exact results, whereas other modes of operation, such as transmitting incomplete batches from node 1 to node 2 along with the normal batches, do not. However, these modes are covered by the approximate model considered in Section 6.

Response time is the sum of the tagged task's sojourn times in each of the two nodes, $S_1 + S_2$ (relabelling the T of the previous section by S_1), which is equivalent to the sum of the sojourn time in the second node and the *reversed* sojourn time in the first node, at equilibrium. This forward-reversed approach to sojourn times is explained in Appendix A and was first used (to

[4] δ_{ij} is the Krönecker delta.

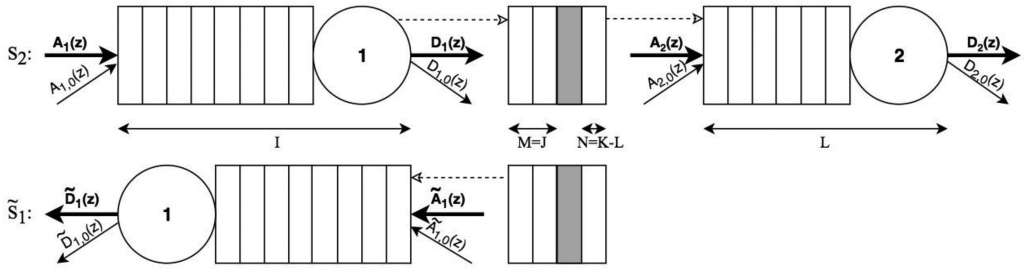

Fig. 4. State at the instant the tagged batch transits from node 1 to node 2 (indicated by the dotted open arrow), in both the forward and reversed processes. The reversed process is shown only for node 1 because sojourn time there is independent of the process at node 2 before the arrival of the tagged batch.

our knowledge) in [1]. Hence, we start measuring time at the instant the tagged task completes service at node 1 and enters node 2. We consider the *middle state* of the pair of nodes at this instant, defined to be the ordered pair comprising the numbers of tasks left behind the tagged task's departure batch at node 1, and in front of it at node 2—i.e., excluding the tagged task's batch in both cases. The idea is to investigate the reversed sojourn time $\tilde{S}_1(I, L, M, N)$ at node 1 and the forward sojourn time $S_2(I, L, M, N)$ at node 2, conditioned on the middle state (I, L) and numbers of tasks behind and in front of the tagged task in its batch, (M, N); refer to Figure 4. The conditional response time random variable is then $\tilde{S}_1(I, L, M, N) + S_2(I, L, M, N)$ and the sought response time distribution can be computed by deconditioning with respect to the middle state and position-in-batch probabilities. In a product-form, tandem batch-network, the middle state's probability distribution is given by the following:

Proposition 4. *In a product-form, tandem batch-network of two nodes at equilibrium:*

(1) The middle state (i, ℓ) has probability equal to $(1 - \rho_1)(1 - \rho_2)\rho_1^i \rho_2^\ell$, where (ρ_1, ρ_2) is the solution vector of the network's rate equations, defined in [14].
(2) The size of the batch of an in-transit tagged task leaving node 1 is independent of the middle state and has ***probability generating function (pgf)*** *$D_1(\rho_1 z)/D_1(\rho_1)$.*
(3) When the tagged task is in a random position in its batch, the joint probability of the numbers of tasks m behind and s in front of it is $\frac{d_{1,m+s+1}\rho_1^{m+s}}{\dot{D}_1(\rho_1)}$.

Proof. The proposition may be proved by "(Conditional) PASTA" [39, 41], but an alternate proof based on flux arguments is given in Appendix C. □

Only normal tasks pass from node 1 to node 2, since special departures from node 1 are discarded from the network. Hence, by definition, the tagged task must be in a normal batch on arrival at node 2. Consequently, we begin by setting the time origin to the instant at which the tagged task arrives at node 1 in a normal batch *in its reversed process* and arrives at node 2 in a normal batch *in its forward process.*

3.1 Non-Overtake-Free Paths at Node 2

When we restrict sojourn times to tasks that remain normal throughout their passage through the two nodes, the network is not overtake-free, as already noted. Consider the sojourn time S (abbreviating S_2) of a given task in the second queue at some arbitrary time, given the following three random variables at that time: the number of tasks in the first queue, I; the number of tasks behind the given task in the second queue, J, which is equal to M at the transition instant shown

in Figure 4; and the number of tasks in front of the given task in the second queue K, so $K = L + N$ in Figure 4. We seek the probability distribution function $\mathbb{P}(S \le t)$, for which we investigate the LST $S^*(\theta) = \mathbb{E}[e^{-\theta S}] = \mathbb{E}[\mathbb{E}[e^{-\theta S} \mid I, J, K]]$.

Proposition 5. *In a tandem pair of minimal discard batch-queues defined by the finite rate-generating functions A_1, D_1, A_2, D_2, with degrees $n_{a1}, n_{d1}, n_{a2}, n_{d2}$, respectively, at equilibrium, the LST $\tau^*_{ijk}(\theta)$ of the probability that a normal task has sojourn time $S \le t$ at node 2* and *remains in a normal batch, given $I = i, J = j, K = k$ at its transition instant between the nodes, has generating functions $G_j(x, z; \theta) = \sum_{i=0}^{\infty} \sum_{k=0}^{\infty} \tau^*_{ijk}(\theta) x^i z^k$, for $j \ge 0$, given by*

$$\mathbb{J}(x, z, \theta)\vec{G}(x, z; \theta) = \vec{E}(x, z; \theta), \tag{6}$$

where the n_{d2}-vector $\vec{G}(x, z; \theta) = (G_0(x, z; \theta), \ldots, G_{n_{d2}-1}(x, z; \theta))$; the $n_{d2} \times n_{d2}$ matrix $\mathbb{J}(x, z, \theta) = (\theta + A_1(1) + A_2(1) + D_1(1) + D_2(1) - A_1(x^{-1}) - D_2(z))\mathbb{I} - \mathbb{M}_1(x) - \mathbb{K}_1(x) - \mathbb{M}_2 - \mathbb{K}_2$; and the $n_{d2} \times n_{d2}$ matrices $\mathbb{M}_1, \mathbb{K}_1, \mathbb{M}_2, \mathbb{K}_2$ have the following non-zero elements (all other elements being 0):

- $\mathbb{M}_1$: $m_{1;j\ell}(x) = d_{1\ell-j} x^{\ell-j}$ *for* $j + 1 \le \ell \le j + n_{d1} \wedge n_{d2} - 1, 0 \le j \le n_{d2} - 1$ *(strictly upper triangular),*
- $\mathbb{K}_1$: $k_{1;j,n_{d2}-1}(x) = \sum_{\ell=n_{d2}}^{j+n_{d1}} d_{1\ell-j} x^{\ell-j}$ *for* $0 \le j \le n_{d2} - 1$ *(only the last column is non-zero),*
- $\mathbb{M}_2$: $m_{2;j\ell} = a_{2,\ell-j}$ *for* $j + 1 \le \ell \le j + n_{a2} \wedge n_{d2} - 1, 0 \le j \le n_{d2} - 1$ *(strictly upper triangular),*
- $\mathbb{K}_2$: $k_{2;j,n_{d2}-1} = \sum_{\ell=n_{d2}}^{j+n_{a2}} a_{2,\ell-j}$ *for* $0 \le j \le n_{d2} - 1$ *(only the last column is non-zero).*

The vector $\vec{E} = (e_j : j = 0, \ldots, n_{d2} - 1)$ is defined by

$$e_j = \sum_{i=1}^{n_{a1}-1} \Big(a_{10i} - x^i \sum_{\ell=i+1}^{n_{a1}} a_{1\ell} x^{-\ell}\Big) g_{ij}(z; \theta) + \left(\frac{D_1(1) - D_1(x)}{1 - x} - A_1(x^{-1}) - A_{10}(1)\right) g_{0j}(z, \theta)$$
$$+ (1 - x)^{-1} \left(\frac{D_2(1) - D_2(z)}{1 - z} - \sum_{\ell=j+2}^{n_{d2}} d_{2\ell} \frac{1 - z^{\ell-j-1}}{1 - z}\right);$$

and the terms $g_{ij}(z; \theta)$ are as yet unknown functions for $0 \le i \le n_{a1} - 1, 0 \le j \le n_{d2} - 1$.

Proof. See Appendix D. □

3.1.1 Determining the Functions $g_{ij}(z; \theta)$ and $G_j(x, z; \theta)$. $\mathbb{J}(x, z)$ is upper triangular and the diagonal elements are all identical except the last one. We denote them by $f_1(x, z)$ and $f_2(x, z)$, respectively, but since we are treating z as a fixed parameter, we abbreviate to $f_1(x)$ and $f_2(x)$. The unknown functions can be determined working row by row, starting at the bottom row.

$G_{n_{d2}-1}(x, z)$ is given by the last row of the matrix $\mathbb{J}$. The last row yields the equation $f_2(x)G_{n_{d2}-1}(x, z) = e_{n_{d2}-1}$ and only $\{g_{i,n_{d2}-1};\ 0 \le i \le n_{a1} - 1\}$ appear on the right-hand side. If $f_2(x)$ has n_{a1} roots within the unit disk, we can determine all the $g_{i,n_{d2}-1}$ and then express $G_{n_{d2}-1}(x, z) = e_{n_{d2}-1}/f_2(x)$ for all x except the roots of $f_2(x)$, where it can be approximated.

$G_k(x, z)$, for $1 \le k \le n_{d2} - 2$, can be computed when all the $G_s(x, z)$ are known for $k < s \le n_{d2} - 1$. The kth row yields $f_1(x)G_k = e_k - (u_1 G_{k+1} + \cdots + u_{n_{d2}-k-1} G_{n_{d2}-1})$, where the known coefficients u_s are from the kth row of $\mathbb{J}$. Again, if $f_1(x)$ has n_{a1} roots inside the unit disk, all the unknown g_{ik} terms $(0 \le i \le n_{a1} - 1)$ in e_k can be determined, and G_k is then easily expressed from the above equation for all x except the roots of $f_1(x)$.

We claim that there are enough roots in the unit disk to determine all the g_{ij} so that G_j $(1 \le j \le n_{d2} - 1)$ can be solved for. The following result proves our claim.

Proposition 6. *$f_1(x) = 0$ and $f_2(x) = 0$ each have n_{a1} roots for x in the unit disk.*

Proof. We appeal to Rouché's theorem.[5] From the definition of $\mathbb{J}$,

$$f_1(x) = A_1(1) + A_2(1) + D_1(1) + D_2(1) - A_1(x^{-1}) - D_2(z) + \theta.$$

Multiplying by $x^{n_{a1}}$, we define $f(x) = [A_1(1) + A_2(1) + D_1(1) + D_2(1) - D_2(z) + \theta]x^{n_{a1}}$ and $g(x) = -A_1(x^{-1})x^{n_{a1}} = \sum_{s=1}^{n_{a1}} -a_{1;s}x^{n_{a1}-s}$, both polynomials in x.

Now, $|f(x)| \geq |A_1(1) + A_2(1) + D_1(1) + D_2(1) + \theta| - |D_2(z)|$ by the triangle inequality and so, for $\mathrm{Re}(\theta) \geq 0$ and $|z| \leq 1$, $|f(x)| \geq A_1(1) + A_2(1) + D_1(1) + D_2(1) - D_2(1) > A_1(1) \geq |g(x)|$ when $|x| \leq 1$. Thus, $f(x) > g(x)$ when $|x| = 1$, and so Rouché's theorem implies that $f_1(x)x^{n_{a1}} = f(x) + g(x)$ has the same number of roots in the unit disk as does $f(x)$, i.e., n_{a1}. Therefore, since $f_1(0) = -a_{1;n_{a1}} \neq 0$, $f_1(x)$ has n_{a1} roots in the unit disk as required.

Similarly, $f_2(x)$ has n_{a1} roots in the unit disk by exactly the same argument. □

3.1.2 LST of Sojourn Time Distribution at Node 2. Let $S_{ms}(t) = P(S \leq t, \Im = 1 | M = m, N = s)$, where M and N are the numbers of tasks behind and in front of the tagged task in its batch, respectively, and $\Im$ is an indicator function indicating that the tagged task leaves node 2 in a normal batch. Both m and s, as well as their sum, are therefore bounded by $n_{d1} - 1$, one less than the maximum batch size of the departure process at the first queue. By the law of total probability, this has LST:

$$\begin{aligned} S^*_{ms}(\theta) &= \sum_{i=0}^{\infty}\sum_{\ell=0}^{\infty}(1-\rho_1)(1-\rho_2)\rho_1^i\rho_2^\ell\tau_{im,s+\ell}(\theta) \\ &= (1-\rho_1)(1-\rho_2)\rho_2^{-s}\sum_{i=0}^{\infty}\sum_{\ell=s}^{\infty}\rho_1^i\rho_2^\ell\tau_{im\ell}(\theta) \\ &= (1-\rho_1)(1-\rho_2)\rho_2^{-s}\left(G_{m\wedge n_{d2}-1}(\rho_1,\rho_2;\theta) - \sum_{\ell=0}^{s-1}\rho_2^\ell\frac{1}{\ell!}\frac{\partial^\ell G_{m\wedge n_{d2}-1}(\rho_1,z;\theta)}{\partial z^\ell}\bigg|_{z=0}\right). \end{aligned}$$

Deconditioning, we find $S^*(\theta) = \sum_{m=0}^{n_{d1}-1}\sum_{s=0}^{n_{d1}-1-m}\phi_{ms}S^*_{ms}(\theta)$. A tagged task departing in a normal batch therefore has (normalized) sojourn time distribution with LST $S^*(\theta)/S^*(0)$.

3.2 Joint Two-Node Sojourn Times

Once we condition on the state (I, J, K) (see Figure 4), the forward and reversed sojourn time random variables are *conditionally* independent—both depend on I. In Proposition 7 we will require a generating function with coefficients that are a sub-sequence of the coefficients in the product of two other generating functions. To deal with this we use the following lemma, which is essentially a "Cauchy inversion formula" that extracts coefficients from a generating function.

Lemma 1.

(1) Let the function $h(z)$ be holomorphic in an annulus $\mathcal{A}$ inside the unit disk of the complex plane and let the circle centred at the origin with radius $r < 1$ lie in $\mathcal{A}$. Then the coefficient of z^n in the Laurent series of $h(z)$ is

$$h_n = \frac{1}{2\pi r^n}\int_0^{2\pi} h(re^{\imath t})e^{-\imath nt}dt$$

for integer n, where $\imath$ is the imaginary unit.

[5] Rouché's theorem states that if f, g are holomorphic functions inside some region $\mathcal{R}$ and $|g| < |f|$ on $\partial\mathcal{R}$, then $f + g$ and f have the same number of roots inside $\mathcal{R}$. In our case $\mathcal{R}$ is the unit disk.

(2) Let $f(z), g(z)$ be holomorphic in the unit disk, with Taylor coefficients $f_0, f_1, \ldots$ and $g_0, g_1, \ldots$, respectively. Then for integer $m \geq 0$,

$$\sum_{j=0}^{\infty} f_j g_{j+m} (re^{\imath\phi})^{2j} = \frac{e^{-2\imath m\phi}}{2\pi r^m} \int_0^{2\pi} f(re^{\imath t}) g(re^{\imath(2\phi - t)}) e^{\imath m t} dt. \tag{7}$$

Proof. Part 1 is routine, using Cauchy's theorem applied to a contour comprising the inner and outer circles of $\mathcal{A}$ and a cut between them in both directions. Then for any contour C in $\mathcal{A}$ that encircles the origin once anticlockwise, $h_n = \frac{1}{2\pi\imath} \oint_C h(z) z^{-n-1} \mathrm{d}z$. Taking C to be the circle with radius r gives the result, which is also well known from Fourier analysis for this particular choice of C.

For part 2, let $h(u) = f(xu)g(y/u)$ for complex variable u and constant complex numbers x, y such that $|y| < |u| < 1/|x|$. Then $h(u)$ is holomorphic in the annulus $\mathcal{A}$ with inner radius $|y|$ and outer radius $1/|x|$, and has Laurent series in $\mathcal{A}$, centered at the origin, $h(u) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} f_i g_j x^i y^j u^{i-j}$. The coefficient of u^{-m} in this series is

$$\sum_{i=0}^{\infty} f_i g_{i+m} x^i y^{i+m} = y^m \sum_{i=0}^{\infty} f_i g_{i+m} (xy)^i.$$

The result now follows by setting $x = 1, y = (re^{\imath\phi})^2$, noting that $r^2 < r < 1$ and using part 1. □

This leads to the following result:

Proposition 7. *An in-transit task with s tasks in front of it and m behind it in its batch has joint sojourn time distribution with LST*

$$T^*_{ms}(\theta_1, \theta_2) = (1 - \rho_1)(1 - \rho_2)\rho_2^{-s} \left[F_{ms}(\rho_1, \rho_2; \theta_1, \theta_2) - \sum_{\ell=0}^{s-1} \frac{\rho_2^\ell}{\ell!} F^{(\ell,z)}_{ms}(\rho_1, 0; \theta_1, \theta_2) \right] \tag{8}$$

for $1 \leq m, s \leq n_{d_1 - 1}$, where for $|z| < 1$ and real r, $0 \leq r < 1$,

$$F_{ms}(r, z; \theta_1, \theta_2) = \frac{1}{2\pi r^{m/2}} \int_0^{2\pi} \tilde{H}_s(\sqrt{r} e^{-\imath t}; \theta_1) G_m(\sqrt{r} e^{\imath t}, z; \theta_2) e^{\imath m t} dt, \tag{9}$$

and, in general, the kth derivative w.r.t. a variable y is denoted by the superscript $\cdot^{(k,y)}$, so that $F^{(\ell,z)}_{ms}(x, z; \theta_1, \theta_2) = \frac{\partial^\ell}{\partial z^\ell} F_{ms}(x, z; \theta_1, \theta_2)$ and $F^{(\ell,z)}_{ms}(\rho_1, 0; \theta_1, \theta_2) \stackrel{def}{=} \frac{\partial^\ell F_{ms}(\rho_1, z; \theta_1, \theta_2)}{\partial z^\ell}|_{z=0}$. The vector function $\vec{\tilde{H}}$ is the reversed process version of $\vec{H}$ at node 1, obtained by using the reversed generating functions $\{\tilde{A}(z), \tilde{D}(z)\}$[6].

[6]The coefficients of these generating functions are the reversed arrival and service rates, routinely calculated by [25].

Proof. Omitting the arguments θ_1, θ_2, where the meaning is clear,

$$\begin{aligned}
T^*_{ms}(\theta_1,\theta_2) &= \sum_{i=0}^{\infty}\sum_{\ell=0}^{\infty} \tilde{Y}^*_{i+m,s}(\theta_1)\tau^*_{im,s+\ell}(\theta_2)(1-\rho_1)\rho_1^i(1-\rho_2)\rho_2^\ell \\
&= \sum_{i=0}^{\infty}\sum_{\ell=s}^{\infty} \tilde{Y}^*_{i+m,s}\tau^*_{im\ell}(1-\rho_1)\rho_1^i(1-\rho_2)\rho_2^{\ell-s} \\
&= (1-\rho_1)(1-\rho_2)\rho_2^{-s}\left[\sum_{i=0}^{\infty}\sum_{\ell=0}^{\infty}\tilde{Y}^*_{i+m,s}\tau^*_{im\ell}\rho_1^i\rho_2^\ell - \sum_{i=0}^{\infty}\sum_{\ell=0}^{s-1}\tilde{Y}^*_{i+m,s}\tau^*_{im\ell}\rho_1^i\rho_2^\ell\right] \\
&= (1-\rho_1)(1-\rho_2)\rho_2^{-s}\left[F_{ms}(\rho_1,\rho_2) - \sum_{\ell=0}^{s-1}\frac{\rho_2^\ell}{\ell!}F^{(\ell,z)}_{ms}(\rho_1,0)\right],
\end{aligned}$$

where $F_{ms}(x,z) = \sum_{i=0}^{\infty}\sum_{\ell=0}^{\infty}\tilde{Y}^*_{i+m,s}\tau^*_{im\ell}x^i z^\ell$ and $F^{(\ell,z)}_{ms}(x,0)/\ell!$ is its coefficient of z^ℓ, namely, $\sum_{i=0}^{\infty}\tilde{Y}^*_{i+m,s}\tau^*_{im\ell}x^i$.

By Lemma 1, with $\phi = 0$, applied to the coefficient of z^{-m},

$$F_{ms}(r^2,z) = \frac{1}{2\pi r^m}\int_0^{2\pi}\tilde{H}_s(re^{-\imath t};\theta_1)G_m(re^{\imath t},z;\theta_2)e^{\imath m t}dt. \tag{10}$$

Substituting $\sqrt{r}$ for r in Equation (10), which is valid because if $r < 1$ then $\sqrt{r} < 1$, yields the result we seek. □

We can now give the main result of the article for the unconditional response time of a normal-to-normal task passing through the two nodes.

Theorem 1. *A normal-to-normal task in a random batch position in the middle state has unconditional response time distribution with LST equal to $T^*(\theta)/T^*(0)$, where*

$$T^*(\theta) = \sum_{s=0}^{n_{d1}-1}\sum_{m=0}^{n_{d1}-1-s}\frac{d_{1,m+s+1}\rho_1^{m+s}T^*_{ms}(\theta,\theta)}{\dot{D}_1(\rho_1)}. \tag{11}$$

Proof. Setting $\theta_1 = \theta_2 = \theta$ to get the LST for the sum of the node sojourn time random variables, the proof is obvious in the light of Propositions 7 and 4. Division by $T^*(0)$, which is the probability that the tagged task departs from node 2 in a full batch and arrives at node 1 in a normal batch, gives the LST corresponding only to normal-to-normal tasks. □

Note that task positions in the transiting batch other than random can be used instead, by alternate choice of ϕ_{ms} to replace $\frac{d_{1,m+s+1}\rho_1^{m+s}}{\dot{D}_1(\rho_1)}$. This gives a simpler result when, for example, the tagged task is first in its batch, corresponding to $s = 0$ only, i.e., $N = 0$ with probability one.

It is encouraging to note that in the special case that $n_{a1} = n_{d1} = n_{a2} = n_{d2} = 1$, the network reduces to a tandem pair of M/M/1 queues, and the reduction of our solution to the M/M/1 case is shown in Appendix E.

3.3 Computing the Remaining Unknown Functions

It remains to determine the functions $F^{(n,z)}_{ms}(x,0;\theta_1,\theta_2)$, which, as in the case already solved when $n = 0$ (in the preceding two subsections), can be obtained from Equation (6), which may be rewritten in the form

$$\left(\mathbb{J}(x,0,\theta) - D_2(z)I\right).\vec{G}(x,z;\theta) = \mathbb{E}(x).\vec{g}(z,\theta) + \vec{e}(x,z), \tag{12}$$

where $\mathbb{E}(x)$ is an $n_{d2} \times n_{a1} n_{d2}$ matrix and $\vec{e}(x,z)$ is an n_{d2}-vector with jth component $e_j(x,z) = \frac{1}{1-x}\left(\frac{D_2(1)-D_2(z)}{1-z} - \sum_{\ell=j+2}^{n_{d2}} d_{2\ell} \frac{1-z^{\ell-j-1}}{1-z}\right)$.

Let $\vec{G}(x,z;\theta)[\vec{e}(x,z) \to \vec{\omega}(x,z)]$ be the solution for $\vec{G}(x,z;\theta)$ (including $\vec{g}(z,\theta)$) of Equation (6), obtained by the method described above, when $\vec{\omega}(x,z)$ is substituted for $\vec{e}(x,z)$.

Proposition 8. *For real r, $0 \le r < 1$,*

$$F_{ms}^{(n,z)}(r,0;\theta_1,\theta_2) = \frac{1}{2\pi r^{m/2}} \int_0^{2\pi} \tilde{H}_s(\sqrt{r}e^{-\imath t};\theta_1)\hat{G}_m^{(n,z)}(\sqrt{r}e^{\imath t},0;\theta_2)e^{\imath m t}dt,$$

where, for $0 \le m \le n_{d2}-1$ and $|x| < 1$,

$$\hat{G}_m^{(n,z)}(x,0;\theta) = G_m(x,0,\theta)\left[\vec{e}(x,0) \to \vec{e}^{(n,z)}(x,0) + \sum_{i=1}^{n \wedge n_{d2}} \binom{n}{i} d_{2i} i!\, \vec{\hat{G}}^{(n-i,z)}(x,0;\theta)\right]. \tag{13}$$

Proof. Differentiating Equation (12) $n \ge 1$ times w.r.t. z using Leibnitz's rule at $z = 0$, and noting that $D_2(0) = 0$, we find for all x in the unit disk,

$$\mathbb{J}(x,0,\theta).\vec{G}^{(n,z)}(x,0;\theta) = \mathbb{E}(x).\vec{g}^{(n,z)}(0,\theta) + \vec{e}^{(n,z)}(x,0) + \sum_{i=1}^{n \wedge n_{d2}} \binom{n}{i} d_{2i} i!\, \vec{G}^{(n-i,z)}(x,0;\theta).$$

Notice that this equation also holds when $n = 0$ by virtue of an empty sum. The mth component of the solution is $G_m^{(n,z)}(x,0;\theta) = \hat{G}_m^{(n,z)}(x,0;\theta)$, as defined in the proposition, and the result follows on setting $x = \sqrt{r}e^{\imath t}$. □

We note that $\{g_{ij}^{(n,z)}(z,\theta) | 0 \le i \le n_{a1}, 0 \le j \le n_{d2}\}$ are also provided as part of the solution of Equation (6) when computing the derivatives; refer to Section 3.1. Equation (13) of Proposition 8 defines a simple recurrence relation on n in $\vec{\hat{G}}^{(n,z)}(x,0;\theta)$, and hence an algorithm for numerical computation. Henceforth, we drop the hats used in the proposition to distinguish algorithmic calculations of derivatives.

4 MOMENTS OF RESPONSE TIME

The results of the previous section give all we need to compute the LST of the response time distribution through the two nodes—or indeed, of the joint sojourn time distribution—and the **probability density function (pdf)** itself can be obtained by numerical Laplace transform inversion. Computation of moments would be highly inaccurate and unstable based on such a numerical estimate of the density function, especially for moments higher than the mean. However, we can compute the moments of the conditional sojourn times at both nodes 1 and 2, by differentiating Equations (4) and (6), respectively, this time w.r.t. θ.

Beginning with node 1, we have (recalling our use of superscripted pairs to denote differentiation):

Proposition 9. *For all x in the unit disk,*

$$\vec{H}^{(p,\theta)}(x;0) = (-1)^p p!(\mathbb{L}(x;0))^{-(p+1)}\mathbb{Y}\mathbb{D}\vec{v}(x).$$

Proof. Equation (4) can be rewritten $\left(\mathbb{L}(x;0) + \theta I\right)\vec{H}(x;\theta) = \mathbb{Y}\mathbb{D}\vec{v}(x)$, where the right-hand side is independent of θ. Differentiating p times w.r.t. θ gives $\mathbb{L}(x;\theta)\vec{H}^{(p,\theta)}(x;\theta) = -p\vec{H}^{(p-1,\theta)}(x;\theta)$. Since we already proved that $\mathbb{L}(x;\theta)$ is non-singular, the result follows at $\theta = 0$. □

For the second node we have the corresponding result:

Proposition 10. *For all x, z in the unit disk and $p > 0$, $\vec{G}^{(p,\theta)}(x,z;0) = \vec{G}(x,z;0)[\vec{e}(x,z) \to -p\vec{G}^{(p-1,\theta)}(x,z;0)]$ and $\vec{G}^{(0,\theta)}(x,z;0) = \vec{G}(x,z;0)$ is the solution of Equation (12) at $\theta = 0$.*

Proof. Proceeding similarly to Section 3.3, we rewrite Equation (6) as

$$\Big(\mathbb{J}(x,z;0) + \theta I\Big).\vec{G}(x,z;\theta) = \mathbb{E}(x).\vec{g}(z;\theta) + \vec{e}(x,z), \tag{14}$$

where $\mathbb{E}(x)$ and $\vec{e}(x,z)$ are as defined before in Section 3.3. Differentiating p times w.r.t. θ then yields

$$\mathbb{J}(x,z;\theta).\vec{G}^{(p,\theta)}(x,z;\theta) = \mathbb{E}(x).\vec{g}^{(p,\theta)}(z;\theta) - p\vec{G}^{(p-1,\theta)}(x,z;\theta).$$

Hence, we have $\vec{G}^{(p,\theta)}(x,z;0) = \vec{G}(x,z;0)[\vec{e} \to \vec{\omega}]$, where $\vec{\omega}(x,z) = -p\vec{G}^{(p-1,\theta)}(x,z;0)$. □

The previous two propositions provide simple iterations for computing pth conditional moments (via the derivatives at $\theta = 0$) for $p = 0, 1, 2, \ldots$, in that order. To find the moments of the unconditioned response time T, we also need the following:

Proposition 11. *For all x in the unit disk, $p > 0$ and $n > 0$,*

$$\vec{G}^{(n,z)(p,\theta)}(x,0;0) = \vec{G}(x,0;0)\left[\vec{e}(x.0) \to -p\vec{G}^{(n,z),(p-1,\theta)}(x,0;0) + \sum_{i=1}^{n\wedge n_{d2}} \binom{n}{i} d_{2i}\, i!\, \vec{G}^{(n-i,z),(p,\theta)}(x,0;0)\right]. \tag{15}$$

Proof. Now we rewrite Equation (6) as

$$\Big(\mathbb{J}(x,0;0) + \theta I - D_2(z)I\Big).\vec{G}(x,z;\theta) = \mathbb{E}(x).\vec{g}(z;\theta) + \vec{e}(x,z).$$

Differentiating $p > 0$ times w.r.t. θ and then n times w.r.t. z at $\theta = z = 0$ gives

$$\mathbb{J}(x,0;0)\vec{G}^{(n,z),(p,\theta)}(x,0;0) = \mathbb{E}(x)g^{(n,z),(p,\theta)}(0;0) - p\vec{G}^{(n,z),(p-1,\theta)}(x,0;0) + \sum_{i=1}^{n\wedge n_{d2}} \binom{n}{i} d_{2i}\, i!\, \vec{G}^{(n-i,z),(p,\theta)}(x,0;0),$$

and the result follows in the now routine way. □

Proposition 11 provides an iterative solution for computing all the $\vec{G}^{(n,z)(p,\theta)}(x,0;0)$ numerically, using the solution method of Section 3.1 for preset values of z with symbolic x. Since $\vec{G}^{(n,z)(p,\theta)}(x,0;0)$ is known for either $n = 0$ or $p = 0$ (from Propositions 8 and 10, respectively), the iteration can proceed in the "alphabetic" ordering of (n,p), *viz.* $(0,0), (0,1), \ldots, (0, n_{d2} - 1), (1,0), (1,1), \ldots, (1, n_{d2} - 1), (2,0), (2,1), \ldots, (2, n_{d2} - 1), \ldots$. All terms on the right-hand side of Equation (15) are then known in each step.

It is now easy to determine the terms $F_{ms}^{(n,z),(p,\theta)}(r,z;0,0)$ that are necessary to find the pth moment of response time by differentiation w.r.t. θ of Equation (7) at $\theta = 0$.

Theorem 2. *For all real $r \in [0,1)$ and z in the unit disk, for $p \geq 0$,*

$$F_{ms}^{(p,\theta)}(r,z;0,0) = \frac{1}{2\pi r^{m/2}} \sum_{i=0}^{p} \binom{p}{i} \int_0^{2\pi} \tilde{H}_s^{(p-i,\theta)}(\sqrt{r}e^{-\imath t};0) G_m^{(i,\theta)}(\sqrt{r}e^{\imath t},z;0) e^{\imath m t}\, dt,$$

and for $n > 0, p > 0$, at $z = 0$,

$$F_{ms}^{(n,z),(p,\theta)}(r,0;0,0) = \frac{1}{2\pi r^{m/2}} \sum_{i=0}^{p} \binom{p}{i} \int_0^{2\pi} \tilde{H}_s^{(p-i,\theta)}(\sqrt{r}e^{-\imath t};0) G_m^{(n,z)(i,\theta)}(\sqrt{r}e^{\imath t},0;0) e^{\imath m t}\, dt,$$

Table 2. Test-Network Specifications and Moments

	Model Parameters		Response Time Moments			
Test Case	Batch Arrivals	Batch Departures	Mean	Std. Dev.	Skewness	Kurtosis
1	{8}, {7}	{12}, {20}	0.45	0.320	1.440	6.145
2	{8}, {7}	{0, 6}, {0, 10}	0.763	0.499	1.390	6.013
3	{8}, {7}	{0, 0, 4}, {0, 1, 6}	1.078	0.673,	1.379	6.011
4	{8}, {7}	{0, 0, 0, 3}, {0, 0, 0, 5}	1.415	0.858	1.363	5.958
5	{2, 3}, {3, 2}	{2, 5}, {6, 7}	0.867	0.574	1.382	5.455
6	{1, 2, 1}, {2, 1, 1}	{1, 2, 1, 1}, {6, 7}	1.183	0.771	1.426	6.250
7	{0, 4}, {7}	{12}, {20}	0.664	0.451	1.417	6.102
8	{0, 0, 0, 2}, {7}	{12}, {20}	1.067	0.697	1.435	6.276
9	{0, $\overset{(5)}{\ldots}$, 0, 1}, {7}	{12}, {20}	1.808	1.158	1.513	6.734

where $\tilde{H}_s^{(p-i,\theta)}(\sqrt{r}e^{-\imath t};0)$, $G_m^{(i,\theta)}(\sqrt{r}e^{\imath t},z;0)$, *and* $G_m^{(n,z)(i,\theta)}(\sqrt{r}e^{\imath t},0;0)$ *are given by Propositions 9, 10, and 11, respectively.*

The p^{th} *moment of response time, conditioned on the batch position of the tagged task, is then*

$$(1-\rho_1)(1-\rho_2)\rho_2^{-s}\left[F_{ms}^{(p,\theta)}(\rho_1,\rho_2;0,0)-\sum_{\ell=0}^{s-1}\frac{\rho_2^\ell}{\ell!}F_{ms}^{(\ell,z),(p,\theta)}(\rho_1,0;0,0)\right], \tag{16}$$

which is deconditioned as in Equation (11).

5 NUMERICS

We consider tandem "test networks" of two batch-queues parameterized by the following sets of arrival and service rates:

The first, "benchmark" network has $M/M/1$ queues and so no special arrivals. Tandem networks of M/M/1 queues exhibit independence at equilibrium by Burke's theorem, so that their response time distribution is a convolution of exponentials with parameters $\mu_1-\lambda_1$ and $\mu_2-\lambda_1-\lambda_2$; see [10, 16], for example. In our case we chose $\lambda_1=8,\lambda_2=7,\mu_1=12,\mu_2=20$. The next three test networks use the same (Poisson) *task rates* but increase the departure batch sizes to show the effect of buffering to make the traffic more bursty. The complete sets of parameters for each test network are given in Table 2 as lists of rates corresponding to batch sizes $1,2,\ldots$ for the external arrival process and for the service process at each node; the "special" arrival rates required for a product-form are not listed, these being given by Proposition 1. The resulting mean, standard deviation, skewness, and kurtosis of response time are also given; these four test networks, 1–4, are in block 1 of the table).

The pdfs of the response times on a path through the two queues of a task in a random position in its batch are shown in Figure 5. We see the increasing spread to the right in the densities as the batch sizes increase, consistent with the moments listed in Table 2 for each case. This is as expected in view of the increasing variableness caused by bursty traffic.

Next, we ran two test networks with batching in both the arrival and service processes, still preserving the task arrival rates: these network specifications and their moments are shown in the second block of Table 2 (test networks 5 and 6). Again, we note in Figure 6 the increasing spread caused by higher batch sizes.

Finally, complementing test networks 1–4, we considered the effect of only increasing the arrival batch size at node 1. As before, we maintained the arriving *task rate* at value 8 and compared

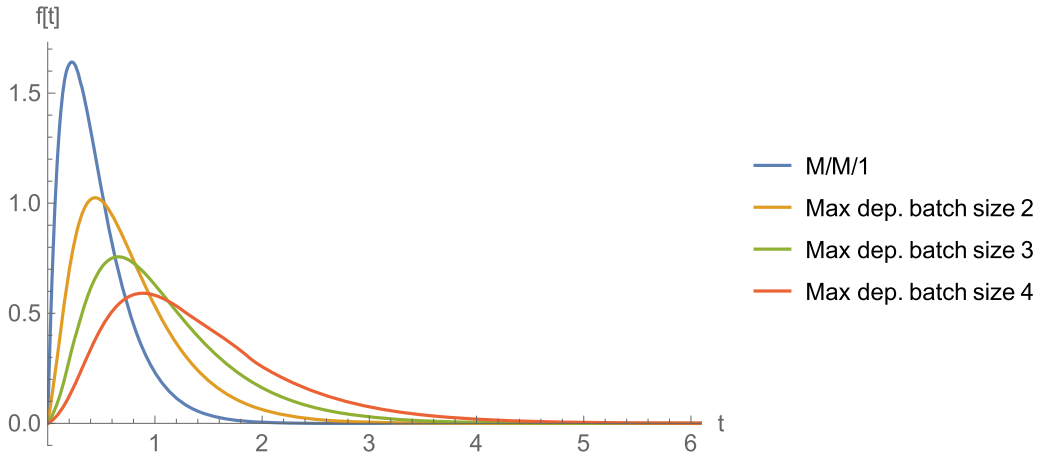

Fig. 5. Density function of the response time in test networks 1–4 showing the effect of increasing departure batch size, where the first queue remains M/M/1.

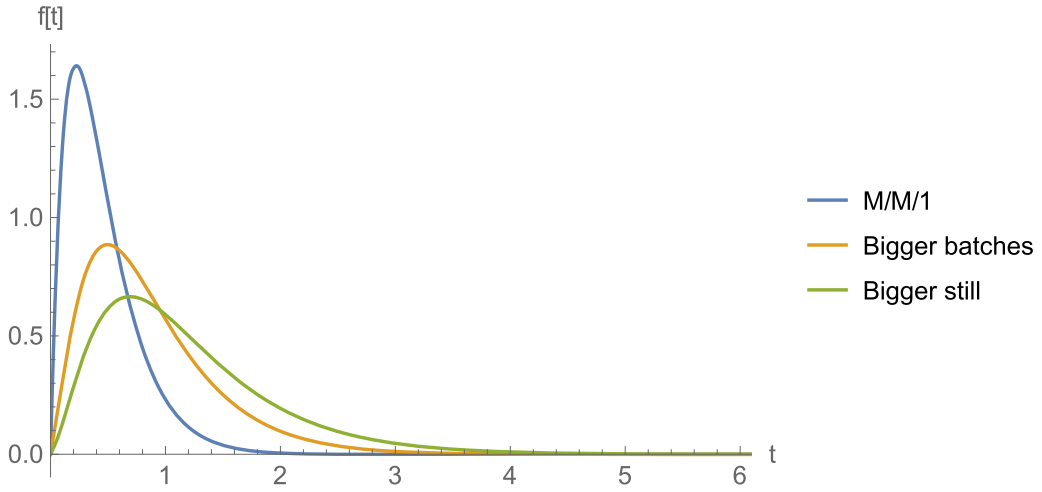

Fig. 6. Density function of the response time in test networks 1, 5, and 6 showing the effect of increasing all batch sizes.

Poisson arrivals of single tasks at rate 8, batches of two tasks at rate 4, batches of four tasks at rate 2, and batches of eight tasks at rate 1. The results can be seen in block 3 of Table 2 (test networks 7–9), as compared with test network 1. The corresponding response time pdfs are shown in Figure 7.

Qualitatively, this chart looks similar to those of Figures 5 and 6. We conclude that larger batch sizes give poorer response times—in terms of both increased mean value and spread—than their smoother unit-batch-size counterparts. This fact is well known in M/M/1, G/M/1, and M/G/1 queues, as well as many more general queuing systems. Table 2 confirms this, but it is interesting to note that skewness and kurtosis are relatively stable across all our network parameterizations. These relate to the heaviness of the tails of a distribution. For response times there is no left-tail,

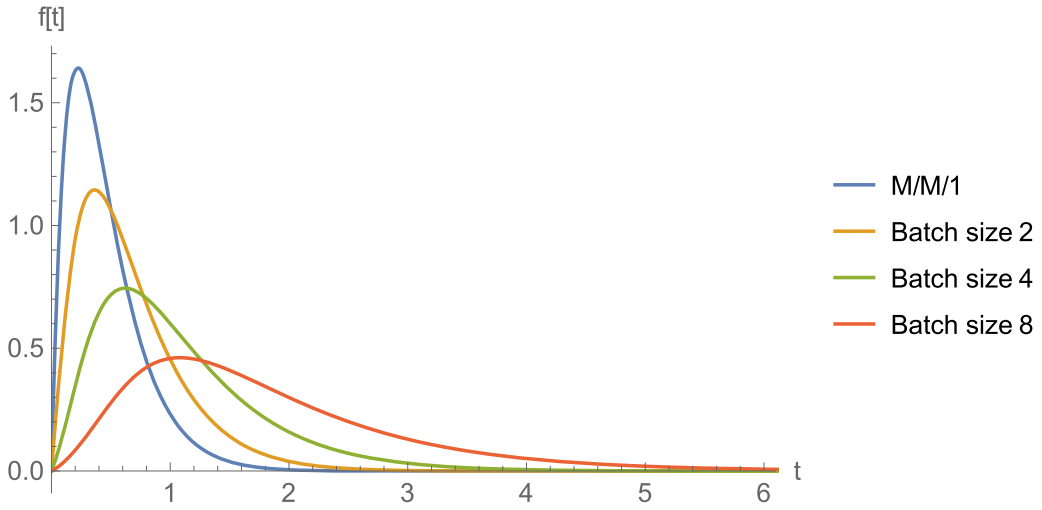

Fig. 7. Density function of the response time in test networks 1 and 7–9 showing the effect of increasing the arriving batch size at node 1 in a tandem pair of M/M/1 queues, where the second queue remains M/M/1.

so skewness is always positive. Since the networks are all Markovian, all state holding times are independent exponential random variables, so response time is a mixture of sums of these and so their tail must also be exponential. This is likely why the skewness and kurtosis do not change very much.

6 RAW BATCH NETWORKS

"Raw" batch networks that do not have additional special arrivals can also be handled by our method using the result of [17] for the non-product-form equilibrium state-probabilities. Partial batches leaving each node may be either discarded from the network, as in [2], or forwarded: either from node 1 to node 2, or leaving node 2 as finished, *incomplete* normal tasks.[7] We consider the former case in this section for compatibility with the rest of the article and to stick to our running application of assembly lines. The following changes from the preceding product-form model are needed:

(1) The special arrival rates are all zero, so that $A_{10}(z) = A_{20}(z) = 0$.
(2) Because the equilibrium queue length probabilities are no longer geometric, the rates in the reversed process at node 1 become state-dependent—at least up to a threshold given by [17]. Therefore, the generating function vector $\vec{\tilde{H}}$ must be recalculated.
(3) More of the $\tilde{\gamma}^*$- and τ^*-terms have to be used directly, i.e., not as part of a generating function.
(4) To handle forwarding of batches, the generating functions $G_j(x, z; \theta)$ and $\tilde{H}_j(x; \theta)$ are easy to modify to reflect the changed state-transitions.

Batches transiting between the nodes still see middle states with probabilities given by PASTA arguments (though now not a product-form) [39, 41], while Propositions 2 and 5 did not require a product-form and so stand (with the aforementioned changes for Proposition 2). Furthermore,

[7] There are also other possibilities considered in the next section.

Proposition 8 can be used to extract coefficients of powers of z and it only remains to find the coefficients of powers of x to obtain all the terms $\tilde{\gamma}^*_{i+m,s}$ and $\tau^*_{im,s+\ell}$ required in a revised expression for $T^*_{ms}(\theta_1, \theta_2)$.

It is shown in [17] that, for large enough integers J, K, the solution for the equilibrium probabilities (when they exist) in the strip $0 \le j \le J, k \ge K$ (called the k-strip) is approximated by the **Spectral Expansion Method (SEM)** of [33] as $(\pi_{0k}, \ldots, \pi_{J,k}) \simeq \sum_{i=1}^{J+1} c_{1,i} \vec{e}_{1,i} x_{1,i}^k$, where $\{x_{1,i} \mid 1 \le i \le J+1\}$ are the roots of a certain polynomial equation with absolute values less than 1 and positive real part, the $\vec{e}_{1,i}$ are eigenvectors corresponding to the $x_{1,i}$, and the $c_{1,i}$ are scalar constants. There is a corresponding result for the j-strip $0 \le k \le K, j \ge J$, namely $(\pi_{j0}, \ldots, \pi_{j,K}) \simeq \sum_{i=1}^{K+1} c_{2,i} \vec{e}_{2,i} x_{2,i}^j$. The probabilities π_{jk} in the region $\{(j,k) \mid 0 \le j < J, 0 \le k < K\}$ are found by direct solution of the balance equations, and finally, at points (j,k) with $j > J, k > K$, $\pi_{jk} \simeq \pi_{JK} \rho_1^{j-J} \rho_2^{k-K}$, as in the product-form considered previously. Numerical application therefore requires first the computation of $J + K + 2$ coefficients for the SEM and a further JK corresponding to the probabilities $\{\pi_{jk} \mid 0 \le j \le J-1, 0 \le k \le K-1\}$. This is done by solving a system of $J + K + 2 + JK$ linear equations (including a normalizing equation) according to the method described in [17].

For one queue alone with finite batches, the SEM is not needed: assuming an equilibrium probability mass function ψ_n for state (queue length) $n \ge 0$, the balance equations for queue lengths less than a threshold Z are solved directly, with states $n \ge Z$ assigned equilibrium probabilities $\psi_Z \rho^{n-Z}$. Then the $Z + 1$ equations are solved together with the normalizing equation. We can consider the reversed process at node 1 of the raw tandem network in this way since it has no special departures, there being no forward special arrivals. Typically, in the tandem network, $Z = J$.

6.1 Response Time Distribution in the Raw Batch Model

The method of the previous sections can now be adapted to the raw batch network (with partial batch discarding) to compute its response time distribution numerically. The generating functions $G_j(x, z; \theta)$ go through unchanged except that the special batch arrival rates are set to 0, i.e., $A_{10}(z) = 0$ (recall there are no special arrivals at the second node while the tagged task is there). The transition rates in the reversed process at node 1 are state-dependent due to the equilibrium probabilities not being geometric below the threshold. However, above the threshold J at node 1, the reversed rates are almost constant due to the approximate product-form. Therefore, once the queue length has become greater than $J + n_{a1} - 1$, the rates *remain* (almost) constant throughout the reversed sojourn time of the tagged task, since after at most $n_{a1} - 1$ service completions the tagged task will have departed (remember that the reversed departure batch size is at most n_{a1}). Therefore, $\tilde{H}_j(x; \theta)$ is the same for all $j \ge J + n_{a1} - 1$.

This extension to two-node, raw, batch tandem networks has been implemented in the easier case[8] that tasks departing node 2 in either partial or full batches are not distinguished; i.e., there is no discarding at node 2 (see [15]). Here, we consider the case with discards at both queues, which is compatible with the assembly-line manufacturing model discussed in Section 2.2 and, more importantly, is consistent with the product-form model for which we obtained exact results. We therefore generalize Proposition 2 as follows (abbreviating $a1, d1$ by a, d, respectively, for consistency of notation as well as brevity).

[8]This case is easier because it *is* overtake-free in the sense defined in Section 2.2. Then the two-node sojourn times are independent.

Proposition 12. *For $0 \le j \le J + n_a - 1$ and any $\kappa > J + n_a$ (where J is the product-form threshold value at node 1), $\tilde{H}_j(x;\theta)$ is given by the recurrence relation:*

$$[D(\rho) + A(\rho^{-1}) - A(x/\rho) + \theta]\tilde{H}_j(x;\theta) - \sum_{s=1}^{n_d} d_s \rho^s \tilde{H}_{j+s}(x;\theta) \approx \sum_{k=0}^{n_a-1} \sum_{s=k+1}^{j+k+1\wedge n_a} \tilde{d}_s(j+k+1)x^k$$
$$- \sum_{k=0}^{\kappa-1} \left[\tilde{A}(j+k+1,1) + \tilde{D}(j+k+1,1) - D(\rho) - A(\rho^{-1})\right] \tilde{\gamma}^*_{kj}(\theta)x^k$$
$$+ \sum_{k=0}^{\kappa-1} x^k \left\{ \sum_{s=1}^{n_d} [\tilde{a}_s(j+k+1) - \tilde{a}_s(\kappa)]\, \tilde{\gamma}^*_{k,j+s}(\theta) + \sum_{s=1}^{n_a} \left[\tilde{d}_s(j+k+s+1) - \tilde{d}_s(\kappa)\right] x^s \tilde{\gamma}^*_{k,j}(\theta) \right\}$$

for $0 \le j \le J + n_a - 1$

$$H_j(x;\theta) = H_{J+n_a-1}(x;\theta) \quad \text{for } j > J + n_a - 1,$$

where the tildes denote the reversed process at node 1 and:

- *n_d and n_a are the maximum batch sizes in the reversed* arrival process *and reversed* departure process*, respectively;*
- *$\tilde{a}_s(i) = d_s\psi_{i+s}/\psi_i$ for $i \ge 0, 1 \le s \le n_d$;*
- *$\tilde{d}_s(i) = a_s\psi_{i-s}/\psi_i$ for $i \ge s, 1 \le s \le n_a$;*
- *$\tilde{A}(i,x) = \sum_{s=1}^{n_d} \tilde{a}_s(i)x^s = \sum_{s=1}^{n_d} d_s\psi_{i+s}/\psi_i x^s$ for $i \ge 0$;*
- *$\tilde{D}(i,x) = \sum_{s=1}^{n_a} \tilde{d}_s(i)x^s = \sum_{s=1}^{i\wedge n_a} a_s\psi_{i-s}/\psi_i x^s$ for $i \ge 0$.*
- *Thus, $\tilde{A}(\kappa,x) \approx \sum_{s=1}^{n_d} d_s(\rho x)^s = D(\rho x)$ and $\tilde{D}(\kappa,x) \approx \sum_{s=1}^{n_a} a_s(x/\rho)^s = A(x/\rho)$ since $\kappa > n_a$.*
- *ψ_i is the marginal probability mass function at equilibrium for node 1, as defined above, i.e., $\psi_i = \sum_{k=0}^{\infty} \pi_{ik}$ for $i \ge 0$, which is known to be asymptotically geometric with parameter ρ, so that $\psi_{i+s}/\psi_i \approx \rho^s$ for $i \ge \kappa - n_a > J$.*

Proof. The proof is given in [17] and repeated in Appendix F for self-containedness. □

As with Proposition 2, this can be expressed in matrix form as $\widetilde{\mathbb{L}}(x;\theta)\vec{\tilde{H}}(x;\theta) = \vec{w}(x,\theta)$, where $\widetilde{\mathbb{L}}(x;\theta) = (D(\rho) + A(\rho^{-1}) - A(x/\rho) + \theta)\mathbb{I} - \tilde{\mathbb{M}} - \tilde{\mathbb{K}}$ as in Equation (4), but here is a $(J + n_a) \times (J + n_a)$ matrix relating to the reversed process in the product-form model, i.e., with $\tilde{m}_{ij} = \rho^{j-i}d_{j-i}$, $i < j \le i + n_d$ (upper triangular) and $\tilde{k}_{i,n_a-1} = \sum_{s=n_a}^{n_d+i} \rho^{s-i}d_{s-i}$ (in the last column of $\tilde{\mathbb{K}}$, the other columns being zero). The jth component (counting $j = 0, \ldots, J + n_a - 1$) of $\vec{w}(x,\theta)$ is (recalling that $\tilde{\gamma}^*_{k,j}(\theta) = \tilde{\gamma}^*_{k,n_a-1}(\theta)$ for $j \ge J + n_a - 1$)

$$w_j = \sum_{k=0}^{n_a-1} \sum_{s=k+1}^{j+k+1\wedge n_a} \tilde{d}_s(j+k+1)x^k - \sum_{k=0}^{\kappa-1} \left[\tilde{A}(j+k+1,1) + \tilde{D}(j+k+1,1) - D(\rho) - A(\rho^{-1})\right] \tilde{\gamma}^*_{kj}(\theta)x^k$$
$$+ \sum_{k=0}^{\kappa-1} x^k \left\{ \sum_{s=1}^{n_d} [\tilde{a}_s(j+k+1) - \tilde{a}_s(\kappa)]\, \tilde{\gamma}^*_{k,j+s\wedge J+n_a-1}(\theta) + \sum_{s=1}^{n_a} \left[\tilde{d}_s(j+k+s+1) - \tilde{d}_s(\kappa)\right] x^s \tilde{\gamma}^*_{k,j}(\theta) \right\},$$

which contains the coefficients $\tilde{\gamma}^*_{k,j}(\theta)$ for $0 \le k \le \kappa - 1, 0 \le j \le J + n_a - 1$. To determine these, we use Equation (22) from the proof of Proposition 12 in Appendix F, *viz:*

$$(\theta + \tilde{A}(j+k+1,1) + \tilde{D}(j+k+1,1))\tilde{\gamma}^*_{kj}(\theta)$$
$$= \sum_{s=1}^{n_d} \tilde{a}_s(j+k+1)\tilde{\gamma}^*_{k,j+s\wedge J+n_a-1}(\theta) + \sum_{s=1}^{k\wedge n_a} \tilde{d}_s(j+k+1)\tilde{\gamma}^*_{k-s,j}(\theta) + \sum_{s=k+1}^{k+j+1\wedge n_a} \tilde{d}_s(j+k+1),$$

starting at $k = 0$ to find the coefficients $\{\tilde{\gamma}^*_{0,j}(\theta) \mid 0 \le j \le J + n_a - 1\}$, then at $k = 1$ to get $\{\tilde{\gamma}^*_{1,j}(\theta) \mid 0 \le j \le J + n_a - 1\}$, and so on up to $\{\tilde{\gamma}^*_{\kappa-1,j}(\theta) \mid 0 \le j \le J + n_a - 1\}$. The vector $\vec{\tilde{H}}(x;\theta)$ can thereby be obtained explicitly as a function of x and the analysis can proceed as in the product-form model.

Theorem 2 goes through largely unchanged, apart from replacing the functions $\tilde{H}_s$, but the spectral expansion may require complex values for the argument x. For this reason, we generalize the function F_{ms} of Equation (9) using Lemma 1, redefining it as

$$F_{ms}(x,z;\theta_1,\theta_2) = \frac{e^{-\imath m\phi}}{2\pi r^{m/2}} \int_0^{2\pi} \tilde{H}_s(\sqrt{r}e^{(\phi-t)\imath};\theta_1) G_m(\sqrt{r}e^{\imath t},z;\theta_2) e^{\imath m t}\,dt \tag{17}$$

for arguments x, z in the complex unit disk, where $x = re^{\imath\phi}$. We will also need derivatives of $F_{ms}(x,z,\theta_1,\theta_2)$ w.r.t. x at $x = 0$ to provide coefficients for use in the lattice strips.

Lemma 2. *For $q \ge 0$ and $1 \le m, s \le n_{d_1} - 1$,*

(1) $F^{(q,x)}_{ms}(0,z,\theta_1,\theta_2) = \frac{1}{(q+m)!}\tilde{H}^{(q+m,x)}_s(0;\theta_1) G^{(q,x)}_m(0,z;\theta_2)$;

(2) $F^{(q,x)(n,z)}_{ms}(0,0,\theta_1,\theta_2) = \frac{1}{(q+m)!}\tilde{H}^{(q+m,x)}_s(0;\theta_1) G^{(q,x),(n,z)}_m(0,0;\theta_2)$.

Proof. For part (1), $\frac{1}{q!}F^{(q,x)}_{ms}(0,z,\theta_1,\theta_2)$ is the coefficient of x^q in the expansion of $F_{ms}(x,z,\theta_1,\theta_2)$, which is, from the proof of Proposition 7, $\sum_{i=0}^{\infty}\sum_{\ell=0}^{\infty}\tilde{\gamma}^*_{i+m,s}\tau^*_{im\ell}x^i z^\ell$. The coefficient is therefore $\tilde{\gamma}^*_{q+m,s}\sum_{\ell=0}^{\infty}\tau^*_{qm\ell}z^\ell$. But $\tilde{H}_s(x;\theta_1)$ and $G_m(x,z;\theta_2)$ are generating functions (in x) of $\tilde{\gamma}^*_{is}$ and $\sum_{\ell=0}^{\infty}\tau^*_{im\ell}z^\ell$, respectively, proving part (1). Part (2) follows by differentiation w.r.t. z n times at $z = 0$. □

Since both $\tilde{H}_s(x;\theta_1)$ and $G_m(x,z;\theta_2)$ are known as expressions in the variable x at each numerical value of z, the derivatives in Lemma 2 are easy to compute by any mathematical software package. We used Wolfram's Mathematica Version 11.3. A bit more efficiently, especially at higher orders of differentiation, recurrence formulas can be derived easily for the derivatives at $x = 0$ using Leibnitz's rule in (revised) Equation (4) and Equation (6), analogously to Proposition 11 and Theorem 2.

Differently from Proposition 7, in the raw batch model $T^*_{ms}(\theta_1,\theta_2)$ is the LST of the probability distribution of the network's joint sojourn times, *jointly with* the numbers of tasks behind and in front of the tagged task in its batch. We first need to find the probability mass function for the position of the tagged task in its batch, which is also different from that given in Proposition 4.

Proposition 13. *At equilibrium, the probability that a randomly chosen task is at position $s + 1$ in a transiting batch of size $m + s + 1$ that sees middle state (j,k) is*[9]

$$\frac{\pi_{j+m+s+1,k}d_{1,m+s+1}}{\sum_{n=1}^{n_{d_1}} n d_{1n}(1-\sum_{i=0}^{n-1}\psi_i)}.$$

Proof. The equilibrium probability that middle state (j,k) is "left behind" by a task departing node 1 in a batch of size n is proportional to the task probability flux from $(j+n,k)$ to $(j,k+n)$[10], that is, $\pi_{j+n,k}d_{1n}n$. Let $\xi_{m,s}$ be the probability that, in each transiting batch of size $m + s + 1$, some chosen task in that batch is at position $s + 1$—i.e., with s tasks in front and m behind. Then $\pi_{j+m+s+1,k}d_{1,m+s+1}(m+s+1)\xi_{m,s}$ is the expected number of chosen tasks in batches of size $m + s + 1$ that see middle state (j,k) in unit time.

[9]The same proposition is used in [17].
[10]Rigorously, by the Ergodic Theorem for Markov chains.

Therefore, the probability that a chosen task is at position $s+1$ in a batch of size $n > s$, and sees middle state (j,k) after its batch's departure, is obtained by normalization as

$$\frac{\pi_{j+n,k} d_{1,n} n \xi_{n-s-1,s}}{\sum_{n'=1}^{n_{d1}} \sum_{j'=0}^{\infty} \sum_{k'=0}^{\infty} \sum_{s'=0}^{n_{d1}-1} \pi_{j'+n',k'} d_{1n'} n' \xi_{n'-s'-1,s'}} = \frac{\pi_{j+n,k} d_{1,n} n \xi_{n-s-1,s}}{\sum_{n'=1}^{n_{d1}} d_{1n'} n' (1 - \sum_{i=0}^{n'-1} \psi_i)}$$

by the Ergodic Theorem for Markov chains. In the case of randomly chosen tasks, $\xi_{n-s-1,s} = 1/n$, giving the desired result. □

The joint probability distribution $T^*_{ms}(\theta_1, \theta_2)$ is now given by the following:

Theorem 3. *In a raw batch network, let $T^*_{ms}(\theta_1, \theta_2)$ be the LST of the probability distribution of the joint node sojourn times, the tagged task leaving in a full batch, and the numbers m and s of tasks behind and in front of the in-transit tagged task in its batch. Then $\tilde{T}^*_{ms}(\theta_1, \theta_2)$ is given by*

$$\begin{aligned}
\eta^{-1}_{ms} \tilde{T}^*_{ms}(\theta_1, \theta_2) &= \pi_{J,K} \rho_1^{m+s+1-J} \rho_2^{-K-s} \\
&\times \left(F_{ms}(\rho_1, \rho_2; \theta_1, \theta_2) - \sum_{k=0}^{K+s} \frac{F^{(0,k)}_{ms}(\rho_1, 0; \theta_1, \theta_2) \rho_2^k}{k!} - \sum_{j=0}^{J-m-s-1} \frac{F^{(j,0)}_{ms}(0, \rho_2; \theta_1, \theta_2) \rho_1^j}{j!} \right) \\
&+ \pi_{J,K} \rho_1^{m+s+1-J} \rho_2^{-K-s} \sum_{j=0}^{J-m-s-1} \sum_{k=0}^{K+s} \rho_1^j \rho_2^k \frac{F^{(j,k)}_{ms}(0,0,\theta_1,\theta_2)}{j!k!} \\
&+ \sum_{i=1}^{J+1} x_{1,i}^{-s} \sum_{j=0}^{J-m-s-1} c_{1,i} e_{1,i;j+m+s+1} \frac{F^{(j,0)}_{ms}(0, x_{1,i}; \theta_1, \theta_2)}{j!} \\
&- \sum_{i=1}^{J+1} x_{1,i}^{-s} \sum_{j=0}^{J-m-s-1} \sum_{k=0}^{K+s-1} c_{1,i} e_{1,i;j+m+s+1} x_{1,i}^k \frac{F^{(j,k)}_{ms}(0,0,\theta_1,\theta_2)}{j!k!} \\
&+ \sum_{i=1}^{K+1} x_{2,i}^{m+s+1} \sum_{k=0}^{K} c_{2,i} e_{2,i;k} \frac{F^{(0,s+k)}_{ms}(x_{2,i}, 0; \theta_1, \theta_2)}{(s+k)!} \\
&- \sum_{i=1}^{K+1} \sum_{k=0}^{K} x_{2,i}^{m+s+1} \sum_{j=0}^{J-m-s-2} c_{2,i} e_{2,i;k} x_{2,i}^j \frac{F^{(j,s+k)}_{ms}(0,0,\theta_1,\theta_2)}{j!(s+k)!} \\
&+ \sum_{j=0}^{J-m-s-2} \sum_{k=0}^{K-1} \pi_{j+m+s+1,k} \frac{F^{(j,s+k)}_{ms}(0,0,\theta_1,\theta_2)}{j!(s+k)!} \\
&- \pi_{J,K} \frac{F^{(J-m-s-1,s+K)}_{ms}(0,0,\theta_1,\theta_2)}{(J-m-s-1)!(s+K)!},
\end{aligned}$$

where

- $\eta_{m,s} = \frac{d_{1,m+s+1}}{\sum_{n'=1}^{n_{d1}} (1 - d_{1n'} \sum_{i=0}^{n'-1} \psi_i)}$,
- $e_{1,i;j}$ *is the jth (counting from 0) component of the vector* $\vec{e}_{1,i}$ ($e_{2,i;k}$ *similarly), and*
- $\pi_{JK} = \sum_{i=1}^{J+1} c_{1,i} \vec{e}_{1,i;J} x_{1,i}^K$.[11]

[11] This quantity is also equal to $\sum_{i=1}^{K+1} c_{2,i} \vec{e}_{2,i;K} x_{2,i}^J$ in an exact model but here the closeness of the values of the two expressions is used as a stopping condition in the algorithm for computing equilibrium state probabilities [17].

Proof. The proof is in Appendix G and basically follows the one given for the simpler model of [17]. The only significant difference is the definition of the function $F_{ms}(x, z; \theta_1, \theta_2)$, which is just a product of generating functions in the simpler model. □

Finally, a normal-to-normal task in a random batch position in the middle state has unconditional response time distribution with LST equal to $T^*(\theta)/T^*(0)$, where

$$T^*(\theta) = \sum_{s=0}^{n_{d1}-1} \sum_{m=0}^{n_{d1}-1-s} T^*_{ms}(\theta, \theta). \tag{18}$$

6.2 Moments

Moments are obtained exactly as in Section 4 except that in the calculation for the reversed process at node 1, the vector $\vec{w}(x, \theta)$ is not constant, so we obtain for $p > 0$ (corresponding to Proposition 9)

$$\vec{\tilde{H}}^{(p,\theta)}(x; 0) = \widetilde{\mathbb{L}}(x; 0)^{-1} \left[\vec{w}^{(p,\theta)}(x, 0) - p\vec{\tilde{H}}^{(p-1,\theta)}(x; 0) \right],$$

where $\vec{w}^{(p,\theta)}(x, 0)$ has jth component (for $0 \le j \le J + n_a - 1$)

$$\begin{aligned} w_j^{(p)} = & -\sum_{k=0}^{\kappa-1} \left[\tilde{A}(j+k+1, 1) + \tilde{D}(j+k+1, 1) - D(\rho) - A(\rho^{-1}) \right] \tilde{\gamma}^{*(p)}_{kj}(0) x^k \\ & + \sum_{k=0}^{\kappa-1} x^k \left\{ \sum_{s=1}^{n_d} \left[\tilde{a}_s(j+k+1) - \tilde{a}_s(\kappa) \right] \tilde{\gamma}^{*(p)}_{k,j+s\wedge J+n_a-1}(0) + \sum_{s=1}^{n_a} \left[\tilde{d}_s(j+k+s+1) - \tilde{d}_s(\kappa) \right] x^s \tilde{\gamma}^{*(p)}_{k,j}(0) \right\}, \end{aligned}$$

and the terms $\gamma^{*(p)}_{kj}(0)$ are obtained by solving directly Equation (22) differentiated p times at $\theta = 0$, *viz.:*

$$\begin{aligned} & (\tilde{A}(j+k+1, 1) + \tilde{D}(j+k+1, 1))\tilde{\gamma}^{*(p)}_{kj}(0) \\ & = \sum_{s=1}^{n_d} \tilde{a}_s(j+k+1)\tilde{\gamma}^{*(p)}_{k,j+s}(0) + \sum_{s=1}^{k\wedge n_a} \tilde{d}_s(j+k+1)\tilde{\gamma}^{*(p)}_{k-s,j}(0) - p\tilde{\gamma}^{*(p-1)}_{kj}(0), \end{aligned} \tag{19}$$

where $\tilde{\gamma}^{*(p-1)}_{kj}(0)$ is known from the previous recursive step. The rest of the calculation mirrors Section 4.

6.3 Numerical Validation

Test cases 5 (arrival and departure batch sizes all equal to 2) and 6 (batch sizes up to 4), detailed in Section 5, were used to parameterize the raw model of the previous section and the results were compared with a regenerative simulation to assess their accuracy [9]. Any errors arise from the truncation point (J, K), above which product-form is assumed to have been attained. We chose truncation point (8, 8) in both cases, based on the criterion described in [17] and good agreement in the equilibrium queue length probabilities. The regenerative simulations each ran for 500,000 regeneration cycles containing 4,708,439 and 5,325,322 task response time records, respectively. Following [4], these sets of cycles were partitioned into batches of size equal to approximately the square root of the number of cycles, i.e., 707, giving a total of 707 batches as well. The pdf of response time was estimated by the relative frequency histogram of each complete set of simulated response times and 95% confidence bands were computed by the batch means method for each point, using the sample variance of each batch's mean value. This gave the charts in Figure 8, where we see excellent agreement with the model's pdfs. The small discrepancy near the mode of the density is most likely due to the approximation introduced by the truncation, but could also come from imprecision in the simulation or the Laplace transform inversion algorithm.

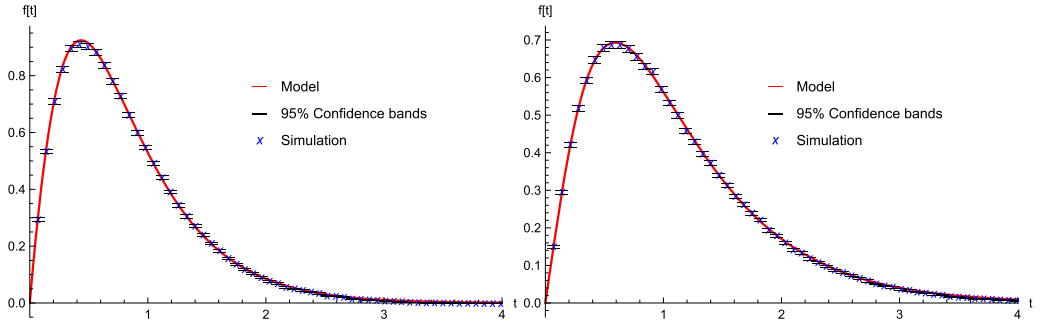

Fig. 8. Density function of the response time in the raw batch network: comparison with simulation. Left chart is for test case 5; right chart is for test case 6.

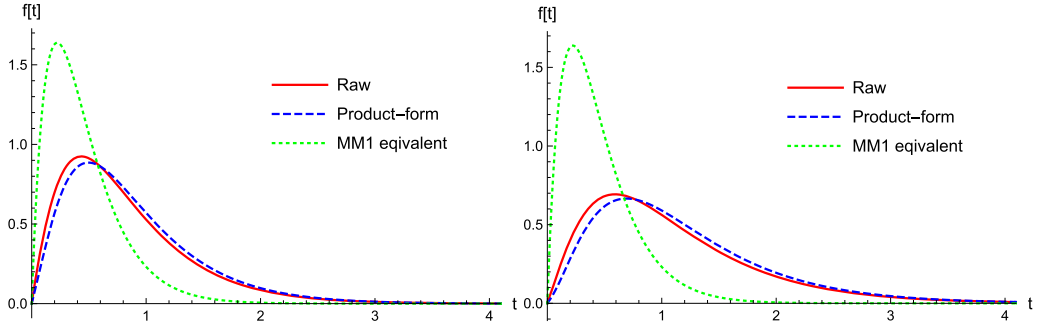

Fig. 9. Density function of the response time in the raw batch network: comparison with the corresponding product-form network (with extra batches) and with the unit-batch network (M/M/1 equivalent) that has the same task throughput and service rates. Left chart is for test case 5; right chart is for test case 6.

The mean response times estimated (with 95% confidence bands) for the two test cases were 0.8151 ± 0.0036 and 1.1009 ± 0.0055, respectively. These values compare exceptionally well with the raw batch model, which gives corresponding means of 0.8134 and 1.1033, well within the confidence intervals.

Finally, we compared the results for the raw batch network with those from the product-form network (Section 5) and with the corresponding network where traffic is "smoothed" into Poisson processes with unit batch sizes and the same task arrival and task-processing rates. The results are shown in Figure 9 and reveal the big qualitative difference between bursty and smooth traffic and also the product-form result not doing too badly, even at a relatively high utilization of over 0.8 at each node; its mean value is about 7% too high. It is the special arrival batches needed by the product-form model that create additional load on the nodes and hence increased mean response time.

7 APPLICATION AREAS

Many systems comprise two phases of service, such as LANs where traffic passes through two routers, e.g., a hub and an expander, as well as many non-computing applications like triage and medical treatment in a hospital. Such have been well modeled by two tandem queues when arrivals are single, but batch arrivals have proved intractable. Previous results on tandem queues, including articles published in this journal in the 1980s such as [1, 5, 38], provided important models for systems like these but required smooth traffic flows that could be well represented by Poisson

processes. The results presented here extend these models to much more realistic situations where traffic is not constrained in this way but can have bounded batches with any distribution of size.

These were obtained first by requiring the often unrealistic assumptions of "special" arrivals at empty queues and "special" departures that clear queues. However, these requirements were removed in the approximate analysis, providing a model with a much wider range of application. Although the special arrivals are simply absent, the special departures may still have a role, defining different semantics in a network context. Any of a number of variants can be modeled, depending on what happens to the special departing batch:

(1) Eject the batch from the network since it is incomplete with respect to the intended batch size—this is the discard model.
(2) Forward the incomplete batch to the next node in the path along which response time is being measured, or as a "finished" partial batch to the external destination.
(3) Send the batch to another node in the larger network of which the path under investigation is a part, e.g., for additional processing—this is equivalent to the discard mode as far as the path is concerned, but not for the overall network semantics.
(4) Avoid the issue entirely by disabling the special batches: i.e., set $d_{ik} = 0$ in states where the queue length at node i is less than k.

These and other variants can all be represented by the approximate models described in Section 6 (or the parallel ones of [15]) because batch sizes have a given maximum value. Therefore, the semi-product-form exists in each variant and the analysis remains the same—the only difference is in the solution provided by the SEM.

In this article, we have considered variant 1, the most general, where the network is not overtake-free since partial batches are discarded as in the manufacturing model of [2]. This kept in mind that simple running example and was also more compatible with the preceding exact analysis. The alternate forwarding mode of transmission, variant 2, is the simplest to analyze because it is overtake-free, and is considered in [15]. The combination of these results facilitates application to far more realistic systems than considered previously in that modern-day bursty traffic can now be accommodated.

7.1 Examples

The intention of the present article is not to explore detailed models for particular case studies, but to provide new theoretical results, showing their domains of practical application. In light of the latter, we might consider central server systems that have product-forms when there are no batches [37]. These paths too have semi-product-form when batches are admitted. This follows because tasks leaving the central server to go to servers other than the second server in the chosen path essentially depart the tandem pair. Then the results on response times can be applied to find quantiles of the time taken for a task to pass along any of the paths through the network.

One instance is that of server farms (or server clusters) [31]. These are often used, for example, to perform large scientific or business calculations, where requests to a scheduling node for a high level of computational resources are distributed to one (or more) of the servers in a "farm." When traffic is bursty, a Markovian central server model can be posed and solved for paths of length two to facilitate response time analysis.

More specifically, consider a web application running on an n-core VM with a single **network interface controller (NIC)**, as described in [3]. A symmetric multicore processor may be modeled by a central server network where each queue ($\text{CPU}_0, \ldots, \text{CPU}_n$) represents a single core. In a typical Linux system, interrupts generated at the NIC are handled by CPU_0, which serves an order of magnitude more interrupts than any other core and we may assume it handles this

task exclusively; the other cores are modeled as M/M/1 queues. When a request arrives from the external network to the VM, the NIC initiates an interrupt at CPU_0, which is processed by triggering the scheduling of a request process at another CPU; after a request has been processed, the response is sent back to the client. A Markovian central-server queueing model of this kind with unit batches—i.e., with Poisson arrivals and exponential service times—has a product-form solution [3]. Hence, the same network with batches has a semi-product-form solution, allowing response time statistics to be computed.

As noted in the introduction, in energy-efficient systems, workload has to be scheduled to introduce *bursts*, with longer idle periods enabling devices to be switched off more often [35]. In the case of consistent, "smooth" traffic—with relatively few long idle periods—switching off the device is rarely possible and increasing the burstiness can improve energy consumption, since idle periods then become longer (for the same throughput), allowing more opportunities for powering down devices. However, this comes at a cost to performance, which is known to deteriorate with increasing burstiness and, ironically, "traffic shaping" has been employed over the years to enhance performance by making traffic *smoother*—i.e., closer to Poisson. There is clearly a trade-off to which the models presented here could contribute.

7.2 Heavy Traffic and Asymptotic Results

Approaching the heavy traffic regime, where the task arrival rates (λ, say) become close to the corresponding maximum task service rates (μ, say), the sub-threshold states are of less interest since queues tend to become very long at equilibrium, where the product-form will hold. However, the threshold values J and K become much higher in heavy traffic, making the spectral expansion expensive; this is still necessary in order to compute the coefficient of geometric terms. Although in heavy traffic a simple approximation, based only on the dominant eigenvalue, is accurate and effective, the computational complexity is still high at $O(\max(J, K)^2)$; see [33].

In very heavy traffic, as $\rho = \lambda/\mu \to 1$, a completely different approach uses stochastic limits in a fluid model [18, 20, 21]. Fluid models replace the countable (or discrete) probability space with a continuous one defined on the real numbers. The arrival processes tend to Brownian motion and, asymptotically, the queues become **Regulated (or Reflected) Brownian motion (RBM)**. The batch-queues considered in this article were analyzed in heavy traffic in precisely this way by Martingale methods in [18]. Other asymptotic results on tandem queues allow for much more general arrival processes. In fact, these processes are only required to have *independent increments*, i.e., need to be Levy processes; arbitrary batches and (independent) inter-arrival times are allowed [20, 21].

8 CONCLUSION

The main contributions of the present work are:

- A closed-form solution for the Laplace transform of response time distribution in a small queuing network with batches was obtained—for the first time to the authors' knowledge. The network was solved using the generating function method, based on the evolution of the underlying Markov process.
- The state-dependent generating functions—expressed as a vector—were defined by a matrix-vector equation, i.e., a *set* of equations, rather than a scalar equation typical in previous studies of this kind.
- The unknown functions in the defining matrix-vector equation were obtained using certain parameter values that made the matrix singular, leading to new linear equations, the number of which was proved equal to the number of unknowns.

- There is one generating function vector for each node and the desired result does not come from just their product in view of the *conditional independence* of the nodes in the non-overtake-free network. Lemma 1 was used to solve this problem by means of a complex integral that picked out the appropriate terms from the product.
- Recurrences were obtained for arbitrary moments of the response time.
- The algorithms can be implemented efficiently—we used Wolfram's Mathematica, Version 12.1—and sample numerical results and graphs were provided.
- Finally, the model was extended to corresponding "raw" batch networks that require neither additional "special" arrival streams nor restrictions on the batch departures; these are considerably more realistic and appropriate for practical applications.

Regarding potential applications, one could argue that the odds of a real-world system satisfying the conditions of a product-form theorem, with extra specific traffic streams, are poor. However, benchmark models that are exact under even contrived parameterizations are certainly of value, especially if they provide reasonable approximations in some cases. Moreover, the excellent approximation for raw batch networks, without product-form and not requiring special streams, is viable practically. The energy-saving scenario considered at the outset of the article needs long idle periods to allow devices to be switched off for significant periods. Unfortunately, in the exact, product-form model, the additional external arrivals occur precisely in the idle states and so would cause the device to be powered up again unnecessarily. The "raw" model avoids this issue and makes it ideal for such systems.

APPENDIX

A FORWARD-REVERSED PROCESS APPROACH TO JOINT SOJOURN TIMES

To illustrate this approach, consider a tandem pair of queues. A possible sample path of a tagged task visiting first node 1 and then node 2 is shown in Figure 10. This task arrives at node 1 seeing three tasks ahead of it, leaves behind a queue of length 4 (including the next task to enter service) on departure, and finds a queue of length 3 just before its arrival at node 2, at this same instant. Therefore, the middle state in this instance is (4,3); the tagged task itself is not counted. The traditional method of analysis investigates only forward sample paths and needs to consider the (transient) probability distribution of the node 2 queue length, conditioned on the state existing just prior to the arrival instant at node 1, to find the probability distribution of the middle state.

Alternatively, we consider joint sample paths in the forward node 2 process and in the reversed node 1 process, *beginning* in a given middle state $\vec{m}$, so $\vec{m} = (4, 3)$ in the sample path shown in Figure 10. For the forward response time at node 2, T_2, we look to the right of the vertical axis, and for the reversed response time at node 1, $\tilde{T}_1$, we look to the left. This approach is motivated by the RCAT methodology, which calculates the reversed generators in synchronizing Markov processes at equilibrium [13]; it was used previously for M/M/1 queues in [1].

Then the joint LST sought for the response time distribution is

$$\mathbb{E}_{(M_1,M_2)}[\tilde{T}_1^*(\theta_1 \mid M_1, M_2)T_2^*(\theta_2 \mid M_1, M_2)],$$

where $\tilde{T}_1^*(\theta_1 \mid M_1, M_2) = \mathbb{E}_{\tilde{T}_1}[e^{-\theta_1 \tilde{T}_1} \mid M_1, M_2]$ and $T_2^*(\theta_2 \mid M_1, M_2) = \mathbb{E}_{T_2}[e^{-\theta_2 T_2} \mid M_1, M_2]$. In an overtake-free network, $\tilde{T}_1$ is independent of M_2 and T_2 is independent of M_1, giving LST of the joint sojourn time $\mathbb{E}_{(M_1,M_2)}[\tilde{T}_1^*(\theta_1 \mid M_1)T_2^*(\theta_2 \mid M_2)]$, which is equal to $\mathbb{E}_{M_1}[\tilde{T}_1^*(\theta_1 \mid M_1)]\mathbb{E}_{M_2}[T_2^*(\theta_2 \mid M_2)]$ if M_1 and M_2 are independent. This is the case for a tandem pair of M/M/1 queues and leads to the result of [1] (which extends to tandem networks of any number of M/M/1 queues).

In our case, $\tilde{T}_1$ is independent of M_2 and $\tilde{T}_1^*(\theta_1 \mid M_1)$ is determined by Proposition 2 applied to the reversed process at node 1, which can be determined for a network at equilibrium quite

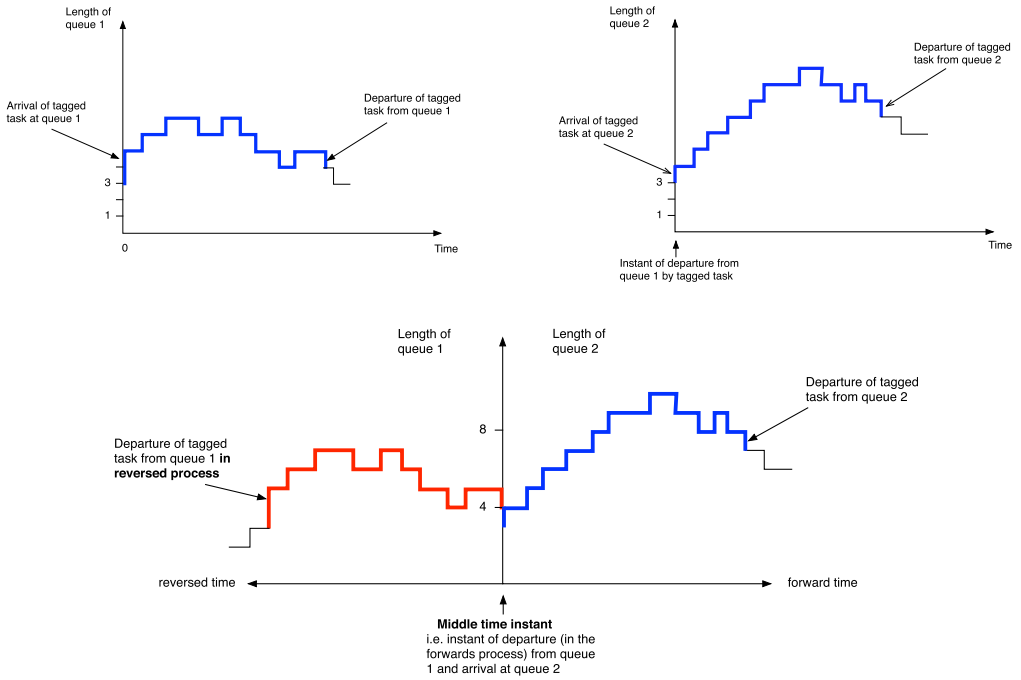

Fig. 10. Upper: Possible sample paths for the queue lengths at each queue during the sojourn of the tagged task. Lower: Forward and reversed sample paths, given middle state (4,3).

simply from the product-form state probabilities. However, T_2 depends on both M_1 and M_2 since the network is not overtake-free, as noted in Section 2.2, and $T_2^*(\theta_2 \mid M_1, M_2)$ is obtained from Proposition 5.

B SPECIAL CASE: MAXIMUM BATCH SIZE 2

The special case with a maximum batch size of two for both arriving and departing batches illustrates the method used in a normal-to-normal batch-queue in a simple way.

Proposition 14. *When $n_a = n_d = 2$, the LST of the sojourn time distribution of a random task in a normal batch (both on arrival and departure) is*

$$T^*(\theta)/T^*(0),$$

where

$$T^*(\theta) = \frac{(1-\rho)[(a_1 + \frac{a_2}{\rho})H_0(\rho;\theta) + a_2 H_1(\rho;\theta) - \frac{a_2}{\rho}H_0(0;\theta)]}{a_1 + 2a_2}$$

and

$$H_0(x;\theta) = \frac{(1-x)d_1^2 + [\theta + a_1 + a_2 + (1-x)(1+2x)d_2]d_1 + [x\theta + (1+x)(a_1+a_2) + x(1-x^2)d_2]d_2}{[\theta + a_1 + a_2 + (1-x)d_1 + (1-x^2)d_2][\theta + (1-x)d_1 + (1-x^2)d_2]}$$

$$H_1(x;\theta) = \frac{d_1 + (1+x)d_2}{\theta + (1-x)d_1 + (1-x^2)d_2}.$$

In addition, the probability that a task arriving in a normal batch departs in a normal batch is $T^*(0)$.

Proof. Using the matrix form of Proposition 2 and omitting θ for brevity, we have

$$[a_1 + a_2 + d_1(1-x) + d_2(1-x^2) + \theta]H_0(x) = a_1 H_1(x) + a_2 H_1(x) + d_1 + d_2 x$$
$$[a_1 + a_2 + d_1(1-x) + d_2(1-x^2) + \theta]H_1(x) = a_1 H_1(x) + a_2 H_1(x) + d_1 + d_2(1+x).$$

Solving these equations gives the desired results for $H_0(x;\theta)$ and $H_1(x;\theta)$.

The LST of the sojourn time distribution then follows as

$$\begin{aligned} T^*(\theta) &= \sum_{m=0}^{n_a-1}\sum_{\ell=0}^{\infty} \gamma^*_{\ell m}(\theta)\pi_{\ell m} = \sum_{m=0}^{1}\sum_{\ell=0}^{\infty} \gamma^*_{\ell m}(\theta)(1-\rho)\sum_{s=0}^{\ell}\phi_{ms}\rho^{\ell-s} \\ &= (1-\rho)\sum_{\ell=0}^{\infty}\left[\gamma^*_{\ell 0}(\theta)(\phi_{00}\rho^{\ell} + \phi_{01}\rho^{\ell-1}\mathbb{I}_{\{\ell\geq 1\}}) + \gamma^*_{\ell 1}(\theta)\phi_{10}\rho^{\ell}\right] \\ &= (1-\rho)\frac{H_0(\rho;\theta)(a_1 + \frac{a_2}{\rho}) + H_1(\rho;\theta)a_2 - H_0(0;\theta)\frac{a_2}{\rho}}{a_1 + 2a_2}. \end{aligned}$$

□

C PROOF OF PROPOSITION 4

Proposition 4. *In a product-form, tandem batch-network of two nodes at equilibrium:*

1. *The middle state (i,ℓ) has probability equal to $(1-\rho_1)(1-\rho_2)\rho_1^i\rho_2^\ell$, where (ρ_1,ρ_2) is the solution vector of the network's rate equations, defined in [14].*
2. *The size of the batch of an in-transit tagged task leaving node 1 is independent of the middle state, and has probability generating function (pgf) $D_1(\rho_1 z)/D_1(\rho_1)$.*
3. *When the tagged task is in a random position in its batch, the joint probability of the numbers of tasks m behind and s in front of it is $\frac{d_{1,m+s+1}\rho_1^{m+s}}{\dot{D}_1(\rho_1)}$.*

Proof. The middle state (i,ℓ) is that existing (without the tagged task's batch) just after a normal departure from node 1. The flux due to normal departures from joint state $(i+n,\ell)$ to $(i,\ell+n)$ is $\pi_{i+n,\ell}d_{1n}$, where n is the size of the batch that passes from node 1 to node 2 and the equilibrium probability mass function $\pi_{i+n,\ell} = (1-\rho_1)(1-\rho_2)\rho_1^{i+n}\rho_2^\ell$ since the network has product-form by Proposition 1. Summing over n, the total flux leading into middle state (i,ℓ), due to normal departures, is $(1-\rho_1)(1-\rho_2)D_1(\rho_1)\rho_1^i\rho_2^\ell$. Hence, the *total* departure flux (into any state) is $D_1(\rho_1)$. The middle state probability is the ratio of the flux into the middle state (i,ℓ) to the total flux, giving part (1).

For part (2), the probability that an in-transit batch has size n is the ratio of the flux into the middle state due to batches of size n and the total flux into the same middle state due to any batch size, i.e., $d_{1n}\rho_1^n/D_1(\rho_1)$, which has the stated pgf.

For part (3), the probability that there are s tasks in front of and m tasks behind a task chosen randomly from a transiting batch is the same as the corresponding probability in the reversed Poisson *arrival* process, i.e., $\frac{d_{1,m+s+1}\rho_1^{m+s+1}}{\rho_1\dot{D}_1(\rho_1)}$, since (s,m) are discrete forwards-backwards recurrence times; e.g., see [16]. □

D PROOF OF PROPOSITION 5

Proposition 5. *In a tandem pair of minimal discard batch-queues defined by the finite rate-generating functions A_1, D_1, A_2, D_2, with degrees $n_{a1}, n_{d1}, n_{a2}, n_{d2}$, respectively, at equilibrium, the LST $\tau^*_{ijk}(\theta)$ of the probability that a normal task has sojourn time $S \leq t$ at node 2* and *remains in*

a normal batch, given $I = i, J = j, K = k$ at its transition instant between the nodes, has generating functions $G_j(x,z;\theta) = \sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\tau^*_{ijk}(\theta)x^i z^k$, for $j \geq 0$, given by

$$\mathbb{J}(x,z,\theta)\vec{G}(x,z;\theta) = \vec{E}(x,z;\theta), \tag{20}$$

where the n_{d2}-vector $\vec{G}(x,z;\theta) = (G_0(x,z;\theta),\ldots,G_{n_{d2}-1}(x,z;\theta))$; the $n_{d2}\times n_{d2}$ matrix $\mathbb{J}(x,z,\theta) = (\theta + A_1(1) + A_2(1) + D_1(1) + D_2(1) - A_1(x^{-1}) - D_2(z))\mathbb{I} - \mathbb{M}_1(x) - \mathbb{K}_1(x) - \mathbb{M}_2 - \mathbb{K}_2$; and the $n_{d2}\times n_{d2}$ matrices $\mathbb{M}_1, \mathbb{K}_1, \mathbb{M}_2, \mathbb{K}_2$ have the following non-zero elements (all other elements being 0):

$\mathbb{M}_1$: $m_{1;j\ell}(x) = d_{1\ell-j}x^{\ell-j}$ *for* $j+1 \leq \ell \leq j + n_{d1} \wedge n_{d2} - 1, 0 \leq j \leq n_{d2}-1$ *(strictly upper triangular),*

$\mathbb{K}_1$: $k_{1;j,n_{d2}-1}(x) = \sum_{\ell=n_{d2}}^{j+n_{d1}} d_{1\ell-j}x^{\ell-j}$ *for* $0 \leq j \leq n_{d2}-1$ *(only the last column is non-zero),*

$\mathbb{M}_2$: $m_{2;j\ell} = a_{2,\ell-j}$ *for* $j+1 \leq \ell \leq j + n_{a2} \wedge n_{d2} - 1, 0 \leq j \leq n_{d2}-1$ *(strictly upper triangular),*

$\mathbb{K}_2$: $k_{2;j,n_{d2}-1} = \sum_{\ell=n_{d2}}^{j+n_{a2}} a_{2,\ell-j}$ *for* $0 \leq j \leq n_{d2}-1$ *(only the last column is non-zero).*

The vector $\vec{E} = (e_j : j = 0,\ldots,n_{d2}-1)$ is defined by

$$e_j = \sum_{i=1}^{n_{a1}-1}\left(a_{10i} - x^i\sum_{\ell=i+1}^{n_{a1}} a_{1\ell}x^{-\ell}\right)g_{ij}(z;\theta) + \left(\frac{D_1(1)-D_1(x)}{1-x} - A_1(x^{-1}) - A_{10}(1)\right)g_{0j}(z,\theta)$$
$$+ (1-x)^{-1}\left(\frac{D_2(1)-D_2(z)}{1-z} - \sum_{\ell=j+2}^{n_{d2}} d_{2\ell}\frac{1-z^{\ell-j-1}}{1-z}\right);$$

and the terms $g_{ij}(z;\theta)$ are as yet unknown functions for $0 \leq i \leq n_{a1}-1, 0 \leq j \leq n_{d2}-1$.

Proof. Let $\tau_{ijk}(t) = \mathbb{P}(S \leq t \mid I = i, J = j, K = k)$ be the probability distribution function of the sojourn time at node 2 of a given normal-to-normal task, conditional on there being i tasks at node 1, j behind at node 2, and k ahead at node 2. Then, since the process describing the network is Markovian, the time evolution of the function τ_{ijk} $(0 \leq i,j,k < \infty)$ can be described by, for small h:

$$\begin{aligned}\tau_{ijk}(t+h) = {} & \Big(1 - h(A_1(1) + A_{10}(1)\epsilon_{i=0} + A_2(1) + D_1(1)\epsilon_{i>0} + D_2(1))\Big)\tau_{ijk}(t)\\
& + h\sum_{\ell=1}^{n_{a1}} a_{1\ell}\tau_{i+\ell,jk}(t) + h\epsilon_{i=0}\sum_{\ell=1}^{n_{a1}-1} a_{10\ell}\tau_{\ell jk}(t) + h\sum_{\ell=1}^{n_{a2}} a_{2\ell}\tau_{i,j+\ell,k}(t)\\
& + h\epsilon_{i>0}\sum_{\ell=1}^{i\wedge n_{d1}} d_{1\ell}\tau_{i-\ell,j+\ell,k}(t) + h\epsilon_{i>0}\sum_{\ell=i+1}^{n_{d1}} d_{1\ell}\tau_{0jk}(t)\\
& + h\sum_{\ell=1}^{k\wedge n_{d2}} d_{2\ell}\tau_{ij,k-\ell}(t) + h\sum_{\ell=k+1}^{j+k+1\wedge n_{d_2}} d_{2\ell}.1 + o(h),\end{aligned}$$

where the degree of the special arrival rate-generating function $A_{10}(z)$ is $n_{a1}-1$, node 1 being a minimal discard batch-queue and $\epsilon_b = 1$ if b is true, 0 if b is false. Note that, at node 2, departures are always possible and there are no special arrivals since the queue is non-empty, the tagged task (at least) being there, by definition. Rearranging, dividing by h, and taking the limit $h \to 0$, we

obtain

$$\dot{\tau}_{ijk}(t) + \Big(A_1(1) + A_{10}(1)\epsilon_{i=0} + A_2(1) + D_1(1)(1-\epsilon_{i=0}) + D_2(1)\Big)\tau_{ijk}(t)$$
$$= \sum_{\ell=1}^{n_{a1}} a_{1\ell}\tau_{i+\ell,jk}(t) + \epsilon_{i=0}\sum_{\ell=1}^{n_{a1}-1} a_{10\ell}\tau_{\ell jk}(t) + \sum_{\ell=1}^{n_{a2}} a_{2\ell}\tau_{i,j+\ell,k}(t)$$
$$+ \sum_{\ell=1}^{i\wedge n_{d1}} d_{1\ell}\tau_{i-\ell,j+\ell,k}(t) + (1-\epsilon_{i=0})\sum_{\ell=i+1}^{n_{d1}} d_{1\ell}\tau_{0jk}(t)$$
$$+ \sum_{\ell=1}^{k\wedge n_{d2}} d_{2\ell}\tau_{ij,k-\ell}(t) + \sum_{\ell=k+1}^{j+k+1\wedge n_{d_2}} d_{2\ell}.$$

Denoting the Laplace transform of $\dot{\tau}_{ijk}(t)$ by $\tau^*_{ijk}(\theta)$, we obtain after routine simplification, using the fact that the Laplace transform of the derivative of a non-negative probability distribution function is the product of the Laplace parameter (θ here) and the Laplace transform of the distribution, and noting that the terms in $D(1)$ cancel when $i = 0$:

$$\Big(\theta + A_1(1) + A_{10}(1)\epsilon_{i=0} + A_2(1) + D_1(1) + D_2(1)\Big)\tau^*_{ijk}(\theta)$$
$$= \sum_{\ell=1}^{n_{a1}} a_{1\ell}\tau^*_{i+\ell,jk}(\theta) + \epsilon_{i=0}\sum_{\ell=1}^{n_{a1}-1} a_{10\ell}\tau^*_{\ell jk}(\theta) + \sum_{\ell=1}^{n_{a2}} a_{2\ell}\tau^*_{i,j+\ell,k}(\theta)$$
$$+ \sum_{\ell=1}^{i\wedge n_{d1}} d_{1\ell}\tau^*_{i-\ell,j+\ell,k}(\theta) + \sum_{\ell=i+1}^{n_{d1}} d_{1\ell}\tau^*_{0jk}(\theta) + \sum_{\ell=1}^{k\wedge n_{d2}} d_{2\ell}\tau^*_{ij,k-\ell}(\theta) + \sum_{\ell=k+1}^{j+k+1\wedge n_{d_2}} d_{2\ell}.$$

The term $\tau_{ijk}(t)$ is the same for all i, k and $j \geq n_{d2} - 1$ because the tagged task is certain to eventually leave node 2 in a full batch (the maximum size of which is n_{d2} including the tagged task) when $J \geq n_{d2} - 1$; the network essentially becomes overtake-free for $j \geq n_{d2} - 1$. Thus, $G_j(x,z) = G_{n_{d2}-1}(x,z)$ for $j \geq n_{d2} - 1$. We now multiply both sides by $x^i z^k$ and sum over $0 \leq i, k < \infty$ for each $j, 0 \leq j \leq n_{d2} - 1$. We consider each term on the right-hand side in turn, dropping the argument θ for brevity. First note that the term $A_{10}(1)\epsilon_{i=0}\tau^*_{ijk}(\theta)$ on the left-hand side gives rise to the term $A_{10}(1)g_{0j}(z,\theta)$ in $e_j(x,z;\theta)$.

First term: Summing over k first and changing the range of i, we obtain

$$\sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{a1}}\sum_{i=\ell}^{\infty} a_{1\ell}\tau^*_{ijk}x^{i-\ell}z^k = \sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{a1}}\sum_{i=0}^{\infty} a_{1\ell}x^{-\ell}\tau^*_{ijk}x^i z^k - \sum_{\ell=1}^{n_{a1}}\sum_{i=0}^{\ell-1} a_{1\ell}x^{i-\ell}\sum_{k=0}^{\infty}\tau^*_{ijk}z^k$$
$$= A_1(x^{-1})G_j(x,z) - \sum_{i=0}^{n_{a1}-1} g_{ij}(z)\sum_{\ell=i+1}^{n_{a1}} a_{1\ell}x^{-(\ell-i)},$$

where $g_{ij}(z) = \sum_{k=0}^{\infty}\tau^*_{ijk}z^k = \frac{1}{i!}\frac{\partial^i G_j(x,z)}{\partial x^i}|_{x=0}$. The second term gives rise to the $n_{a1} \times n_{d2}$ unknown functions of z on the right-hand side of Equation (20), which will be determined using the analyticity of $G(x,z)$ for $|x|, |z| < 1$ below.

Second term: The sum over i yields only the first term (for $i = 0$), so we obtain

$$\sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{a1}-1} a_{10\ell}\tau^*_{\ell jk}z^k = \sum_{i=1}^{n_{a1}-1} a_{10i}g_{ij}(z).$$

This expression combines with the previous one on the right-hand side of Equation (20).

Third term: Straightforwardly, we get

$$\sum_{\ell=1}^{n_{a2}} a_{2\ell} G_{j+\ell\wedge n_{d2}-1}(x,z) = \sum_{\ell=j+1}^{j+n_{a2}} a_{2,\ell-j} G_{\ell\wedge n_{d2}-1}(x,z)$$
$$= \sum_{\ell=j+1}^{j+n_{a2}\wedge n_{d2}-1} a_{2,\ell-j} G_{\ell}(x,z) + \sum_{\ell=n_{d2}}^{j+n_{a2}} a_{2,\ell-j} G_{n_{d2}-1}(x,z).$$

These terms give rise to the matrices M_2 and K_2, respectively, on the left-hand side of Equation (20).

Fourth term: Changing the summation domains of ℓ and i, we have

$$\sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{d1}}\sum_{i=\ell}^{\infty} d_{1\ell}\tau^*_{i-\ell,j+\ell,k}x^i z^k = \sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{d1}}\sum_{i=0}^{\infty} d_{1\ell}x^\ell\tau^*_{i,j+\ell,k}x^i z^k = \sum_{\ell=j+1}^{j+n_{d1}} d_{1\ell-j}x^{\ell-j}G_{\ell\wedge n_{d2}-1}(x,z)$$
$$= \sum_{\ell=j+1}^{j+n_{d1}\wedge n_{d2}-1} d_{1\ell-j}x^{\ell-j}G_{\ell}(x,z) + \sum_{\ell=n_{d2}}^{j+n_{d1}} d_{1\ell-j}x^{\ell-j}G_{n_{d2}-1}(x,z).$$

These terms give rise to the matrices M_1 and K_1, respectively, on the left-hand side of Equation (20).

Fifth term: Again, changing the summation domains of ℓ and i, we get

$$\sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{d1}}\sum_{i=0}^{\ell-1} x^i d_{1\ell}\tau^*_{0jk}z^k = \sum_{k=0}^{\infty}\sum_{\ell=1}^{n_{d1}} d_{1\ell}\tau^*_{0jk}z^k(1-x^\ell)/(1-x) = \left(\frac{D_1(1)-D_1(x)}{1-x}\right)g_{0j}(z).$$

This term appears on the right-hand side of Equation (20).

Sixth term: This time, changing the summation domains of ℓ and k yields

$$\sum_{i=0}^{\infty}\sum_{\ell=1}^{n_{d2}}\sum_{k=\ell}^{\infty} d_{2\ell}\tau^*_{ij,k-\ell}x^i z^k = \sum_{i=0}^{\infty}\sum_{\ell=1}^{n_{d2}}\sum_{k=0}^{\infty} d_{2\ell}z^\ell\tau^*_{ijk}x^i z^k = D_2(z)G_j(x,z).$$

This term appears on the left-hand side of Equation (20).

Seventh term: Here, the sum over i gives a factor $1/(1-x)$ and we get

$$(1-x)^{-1}\sum_{k=0}^{\infty}\sum_{\ell=k+1}^{j+k+1\wedge n_{d_2}} d_{2\ell}z^k = (1-x)^{-1}\sum_{\ell=1}^{n_{d2}} d_{2\ell}\sum_{k=\ell-j-1\vee 0}^{\ell-1} z^k$$
$$= (1-x)^{-1}\sum_{\ell=1}^{n_{d2}} d_{2\ell}\left((1-z^\ell)/(1-z) - \sum_{k=0}^{\ell-j-2} z^k\right)$$
$$= (1-x)^{-1}\left(\frac{D_2(1)-D_2(z)}{1-z} - \sum_{\ell=j+2}^{n_{d2}} d_{2\ell}\frac{1-z^{\ell-j-1}}{1-z}\right),$$

where $\vee$ is the infix operator for the maximum of two numbers.[12] This gives the final terms on the right-hand side of Equation (20), with no unknowns.

□

[12] $\vee$ has the same (lowest) operator precedence as $\wedge$.

E JOINT SOJOURN TIMES IN THE TWO-NODE SPECIAL M/M/1 CASE

Consider the case where both nodes are M/M/1 queues with arrival and departure rates $\lambda_1, \lambda_2, \mu_1, \mu_2$, respectively. Then, the reversed rates of the first queue are $\mu_1\rho_1$ and $\lambda_1\rho_1^{-1}$; see [16, 25], for example. All batches have size one, hence $n_{a1} = n_{a2} = n_{d1} = n_{d2} = 1$, and so there are no special arrivals or departures. The matrix form Equation (4) yields for the reversed process at node 1:

$$(\mu_1\rho_1 + \lambda_1\rho_1^{-1} - \lambda_1\rho_1^{-1}x + \theta - \mu_1\rho_1)H_0(x;\theta) = \lambda_1\rho_1^{-1}$$
$$H_0(x;\theta) = \frac{\lambda_1\rho_1^{-1}}{[\lambda_1\rho_1^{-1}(1-x) + \theta]}.$$

Therefore, evaluating the generating function at $\rho_1 = \lambda_1/\mu_1$ and multiplying by $(1-\rho_1)$, we get

$$\widetilde{S}_1^*(\theta) = (1-\rho_1)H_0(\rho_1;\theta) = (1-\rho_1)\frac{\lambda_1\rho_1^{-1}}{(\lambda_1\rho_1^{-1}(1-\rho_1) + \theta)} = \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 + \theta}. \tag{21}$$

By Proposition 5, the forward sojourn time at the second queue is given by

$$\left[\lambda_1 + \lambda_2 + \mu_1 + \mu_2 - \frac{\lambda_1}{x} - \mu_2 z + \theta - \mu_1 x - \lambda_2\right] G_0(x,z;\theta) = \left[\mu_1 - \frac{\lambda_1}{x}\right] G_0(0,z;\theta) + \frac{\mu_2}{1-x}.$$

Let x_0 be the value of x at which the coefficient of $G_0(x,z;\theta)$ on the left-hand side is zero, i.e. $\lambda_1 + \mu_1 + \mu_2 - \frac{\lambda_1}{x_0} - \mu_2 z + \theta - \mu_1 x_0 = 0$. Then, equating the right-hand side to zero,

$$G_0(0,z;\theta) = \frac{\mu_2}{x_0 - 1}\frac{1}{\left[\mu_1 - \frac{\lambda_1}{x_0}\right]} = \frac{\mu_2}{x_0\mu_1 - \mu_1 - \lambda_1 + \frac{\lambda_1}{x_0}} = \frac{\mu_2}{\mu_2(1-z) + \theta}.$$

Therefore, at $x \neq 0$,

$$\begin{aligned} G_0(x,z;\theta) &= \frac{\mu_2[(\mu_1 - \frac{\lambda_1}{x})(1-x) + \theta + \mu_2(1-z)]}{(1-x)[\theta + \mu_2(1-z)]}\frac{1}{(\lambda_1 + \mu_1 + \mu_2 - \frac{\lambda_1}{x} - \mu_2 z + \theta - \mu_1 x)} \\ &= \frac{\mu_2}{(1-x)[\theta + \mu_2(1-z)]}. \end{aligned}$$

Hence, the LST of the sojourn time distribution at the second queue is

$$S_2^*(\theta) = (1-\rho_1)(1-\rho_2)G_0(\rho_1,\rho_2;\theta) = \frac{\mu_2(1-\rho_2)}{\theta + \mu_2(1-\rho_2)},$$

which is precisely what we get by ignoring the first queue and just using $G_0(0,z;\theta)$. The sojourn times are therefore independent, as is well known, and the pgf of their sum is the product $\widetilde{S}_1^*(\theta)S_2^*(\theta)$.

Proceeding directly via Equation (8) and keeping in mind that $\tau_{ij\ell}$ does not depend on i, the required LST is

$$\begin{aligned} (1-\rho_1)(1-\rho_2)F_{00}^*(\theta) &= (1-\rho_1)(1-\rho_2)\sum_{i=0}^{\infty}\sum_{\ell=0}^{\infty}\tilde{\gamma}_{i0}\tau_{i0\ell}\rho_1^i\rho_2^\ell \\ &= (1-\rho_1)(1-\rho_2)H_0(\rho_1;\theta)G_0(0,\rho_2;\theta) \\ &= \frac{\mu_1 - \lambda_1}{\theta + \mu_1 - \lambda_1}\frac{\mu_2(1-\rho_2)}{\theta + \mu_2(1-\rho_2)}, \end{aligned}$$

as expected.

F PROOF OF PROPOSITION 12

Proposition 12. *For $0 \le j \le J + n_a - 1$ and any $\kappa > J + n_a$ (where J is the product-form threshold value at node 1), $\tilde{H}_j(x;\theta)$ is given by the recurrence relation:*

$$[D(\rho) + A(\rho^{-1}) - A(x/\rho) + \theta]\tilde{H}_j(x;\theta) - \sum_{s=1}^{n_d} d_s \rho^s \tilde{H}_{j+s}(x;\theta) \approx \sum_{k=0}^{n_a-1} \sum_{s=k+1}^{j+k+1\wedge n_a} \tilde{d}_s(j+k+1)x^k$$
$$- \sum_{k=0}^{\kappa-1} \left[\tilde{A}(j+k+1,1) + \tilde{D}(j+k+1,1) - D(\rho) - A(\rho^{-1})\right] \tilde{\gamma}^*_{kj}(\theta)x^k$$
$$+ \sum_{k=0}^{\kappa-1} x^k \left\{ \sum_{s=1}^{n_d} [\tilde{a}_s(j+k+1) - \tilde{a}_s(\kappa)]\, \tilde{\gamma}^*_{k,j+s}(\theta) + \sum_{s=1}^{n_a} \left[\tilde{d}_s(j+k+s+1) - \tilde{d}_s(\kappa)\right] x^s \tilde{\gamma}^*_{k,j}(\theta) \right\}$$

for $0 \le j \le J + n_a - 1$

$H_j(x;\theta) = H_{J+n_a-1}(x;\theta)$ *for* $j > J + n_a - 1$,

where the tildes denote the reversed process at node 1 and:

- *n_d and n_a are the maximum batch sizes in the reversed* arrival process *and reversed* departure process*, respectively;*
- *$\tilde{a}_s(i) = d_s\psi_{i+s}/\psi_i$ for $i \ge 0, 1 \le s \le n_d$;*
- *$\tilde{d}_s(i) = a_s\psi_{i-s}/\psi_i$ for $i \ge s, 1 \le s \le n_a$;*
- *$\tilde{A}(i,x) = \sum_{s=1}^{n_d} \tilde{a}_s(i)x^s = \sum_{s=1}^{n_d} d_s\psi_{i+s}/\psi_i x^s$ for $i \ge 0$;*
- *$\tilde{D}(i,x) = \sum_{s=1}^{n_a} \tilde{d}_s(i)x^s = \sum_{s=1}^{i\wedge n_a} a_s\psi_{i-s}/\psi_i x^s$ for $i \ge 0$.*
- *Thus, $\tilde{A}(\kappa,x) \approx \sum_{s=1}^{n_d} d_s(\rho x)^s = D(\rho x)$ and $\tilde{D}(\kappa,x) \approx \sum_{s=1}^{n_a} a_s(x/\rho)^s = A(x/\rho)$ since $\kappa > n_a$.*
- *ψ_i is the marginal probability mass function at equilibrium for node 1, as defined above, i.e., $\psi_i = \sum_{k=0}^{\infty} \pi_{ik}$ for $i \ge 0$, which is known to be asymptotically geometric with parameter ρ, so that $\psi_{i+s}/\psi_i \approx \rho^s$ for $i \ge \kappa - n_a > J$.*

Proof. We parallel the proof of Proposition 2, adapting it to deal with state-dependent rates. In an infinitesimal interval $(t, t+h]$, we have, for $k, j \ge 0$:

$$\tilde{\gamma}_{kj}(t+h) = h\sum_{s=1}^{n_d} \tilde{a}_s(j+k+1)\tilde{\gamma}_{k,j+s}(t) + h\sum_{s=1}^{k\wedge n_a} \tilde{d}_s(j+k+1)\tilde{\gamma}_{k-s,j}(t) + h\sum_{s=k+1}^{k+j+1\wedge n_a} \tilde{d}_s(j+k+1)$$
$$+ [1 - h\tilde{A}(j+k+1,1) - h\tilde{D}(j+k+1,1)]\tilde{\gamma}_{kj}(t) + o(h).$$

Notice that there are no losses in the reversed process at node 1 since there are no special arrivals in the forward process. This is reflected in the definition of $\tilde{D}(i,x)$.

After rearranging, dividing by h, and taking the limit $h \to 0$, we take the LST of both sides to obtain

$$(\theta + \tilde{A}(j+k+1,1) + \tilde{D}(j+k+1,1))\tilde{\gamma}^*_{kj}(\theta)$$
$$= \sum_{s=1}^{n_d} \tilde{a}_s(j+k+1)\tilde{\gamma}^*_{k,j+s}(\theta) + \sum_{s=1}^{k\wedge n_a} \tilde{d}_s(j+k+1)\tilde{\gamma}^*_{k-s,j}(\theta) + \sum_{s=k+1}^{k+j+1\wedge n_a} \tilde{d}_s(j+k+1). \qquad (22)$$

Multiplying by x^k and then summing from $k=0$ to κ and from $k=\kappa$ to infinity, we get

$$\begin{aligned}
&\sum_{k=0}^{\kappa-1}\left[\tilde{A}(j+k+1,1)+\tilde{D}(j+k+1,1)-\tilde{A}(\kappa,1)-\tilde{D}(\kappa,1)\right]\tilde{\gamma}^*_{kj}(\theta)x^k+(\theta+\tilde{A}(\kappa,1)+\tilde{D}(\kappa,1))\tilde{H}_j(x;\theta)\\
&=\sum_{k=0}^{\infty}x^k\sum_{s=k+1}^{k+j+1\wedge n_a}\tilde{d}_s(j+k+1)+\sum_{s=1}^{n_d}\sum_{k=0}^{\infty}\tilde{a}_s(\kappa)\tilde{\gamma}^*_{k,j+s}(\theta)x^k+\sum_{s=1}^{n_d}\sum_{k=0}^{\kappa-1}\left[\tilde{a}_s(j+k+1)-\tilde{a}_s(\kappa)\right]\tilde{\gamma}^*_{k,j+s}(\theta)x^k\\
&\quad+\sum_{s=1}^{n_a}\sum_{k=0}^{\infty}\tilde{d}_s(\kappa)x^s\tilde{\gamma}^*_{k,j}(\theta)x^k+\sum_{s=1}^{n_a}\sum_{k=0}^{\kappa-1}\left[\tilde{d}_s(j+k+s+1)-\tilde{d}_s(\kappa)\right]x^s\tilde{\gamma}^*_{k,j}(\theta)x^k\\
&=\sum_{s=1}^{n_d}\tilde{a}_s(\kappa)\tilde{H}_{j+s}(x;\theta)+\tilde{D}(\kappa,x)\tilde{H}_j(x;\theta)+\sum_{k=0}^{n_a-1}\sum_{s=k+1}^{k+j+1\wedge n_a}\tilde{d}_s(j+k+1)x^k\\
&\quad+\sum_{k=0}^{\kappa-1}x^k\left\{\sum_{s=1}^{n_d}\left[\tilde{a}_s(j+k+1)-\tilde{a}_s(\kappa)\right]\tilde{\gamma}^*_{k,j+s}(\theta)+\sum_{s=1}^{n_a}\left[\tilde{d}_s(j+k+s+1)-\tilde{d}_s(\kappa)\right]x^s\tilde{\gamma}^*_{k,j}(\theta)\right\}. \qquad \square
\end{aligned}$$

G PROOF OF THEOREM 3

Theorem 3. *In a raw batch network, let $T^*_{ms}(\theta_1,\theta_2)$ be the LST of the probability distribution of the joint node sojourn times, the tagged task leaving in a full batch, and the numbers m and s of tasks behind and in front of the in-transit tagged task in its batch. Then $\tilde{T}^*_{ms}(\theta_1,\theta_2)$ is given by*

$$\begin{aligned}
\eta^{-1}_{ms}\tilde{T}^*_{ms}(\theta_1,\theta_2)&=\pi_{J,K}\rho_1^{m+s+1-J}\rho_2^{-K-s}\\
&\times\left(F_{ms}(\rho_1,\rho_2;\theta_1,\theta_2)-\sum_{k=0}^{K+s}\frac{F^{(0,k)}_{ms}(\rho_1,0;\theta_1,\theta_2)\rho_2^k}{k!}-\sum_{j=0}^{J-m-s-1}\frac{F^{(j,0)}_{ms}(0,\rho_2;\theta_1,\theta_2)\rho_1^j}{j!}\right)\\
&+\pi_{J,K}\rho_1^{m+s+1-J}\rho_2^{-K-s}\sum_{j=0}^{J-m-s-1}\sum_{k=0}^{K+s}\rho_1^j\rho_2^k\frac{F^{(j,k)}_{ms}(0,0,\theta_1,\theta_2)}{j!k!}\\
&+\sum_{i=1}^{J+1}x_{1,i}^{-s}\sum_{j=0}^{J-m-s-1}c_{1,i}e_{1,i;j+m+s+1}\frac{F^{(j,0)}_{ms}(0,x_{1,i};\theta_1,\theta_2)}{j!}\\
&-\sum_{i=1}^{J+1}x_{1,i}^{-s}\sum_{j=0}^{J-m-s-1}\sum_{k=0}^{K+s-1}c_{1,i}e_{1,i;j+m+s+1}x_{1,i}^k\frac{F^{(j,k)}_{ms}(0,0,\theta_1,\theta_2)}{j!k!}\\
&+\sum_{i=1}^{K+1}x_{2,i}^{m+s+1}\sum_{k=0}^{K}c_{2,i}e_{2,i;k}\frac{F^{(0,s+k)}_{ms}(x_{2,i},0;\theta_1,\theta_2)}{(s+k)!}\\
&-\sum_{i=1}^{K+1}\sum_{k=0}^{K}x_{2,i}^{m+s+1}\sum_{j=0}^{J-m-s-2}c_{2,i}e_{2,i;k}x_{2,i}^j\frac{F^{(j,s+k)}_{ms}(0,0,\theta_1,\theta_2)}{j!(s+k)!}\\
&+\sum_{j=0}^{J-m-s-2}\sum_{k=0}^{K-1}\pi_{j+m+s+1,k}\frac{F^{(j,s+k)}_{ms}(0,0,\theta_1,\theta_2)}{j!(s+k)!}\\
&-\pi_{J,K}\frac{F^{(J-m-s-1,s+K)}_{ms}(0,0,\theta_1,\theta_2)}{(J-m-s-1)!(s+K)!},
\end{aligned}$$

where

- $\eta_{m,s} = \frac{d_{1,m+s+1}}{\sum_{n'=1}^{n_{d1}}(1-d_{1n'}\sum_{i=0}^{n'-1}\psi_i)}$.
- $e_{1,i;j}$ *is the* j*th (counting from 0) component of the vector* $\vec{e}_{1,i}$ ($e_{2,i;k}$ *similarly).*
- $\pi_{JK} = \sum_{i=1}^{J+1} c_{1,i}\vec{e}_{1,i;J}x_{1,i}^{K}$.[13]

Proof. Given m tasks behind and s tasks in front of the tagged task in its middle state, we have

$$\tilde{T}^*_{ms}(\theta_1,\theta_2) = \sum_{j=0}^{\infty}\sum_{k=0}^{\infty}\eta_{ms}\pi_{j+m+s+1,k}\tilde{Y}^*_{j+m,s}(\theta_1)\tau^*_{s+k}(\theta_2).$$

Partitioning the sum according to the spectral solution and noting that $J, K > n_{d1}$ and hence $J, K > m+s$,

$$\begin{aligned}
&\eta_{ms}^{-1}\tilde{T}^*_{ms}(\theta_1,\theta_2)\\
&= \left(\sum_{j=J-m-s}^{\infty}\sum_{k=K+1}^{\infty} + \sum_{j=0}^{J-m-s-1}\sum_{k=K}^{\infty} + \sum_{j=J-m-s-1}^{\infty}\sum_{k=0}^{K} + \sum_{j=0}^{J-m-s-2}\sum_{k=0}^{K-1}\right)\pi_{j+m+s+1,k}\tilde{Y}^*_{j+m,s}(\theta_1)\tau^*_{s+k}(\theta_2)\\
&\quad - \pi_{J,K}\tilde{Y}^*_{J-s-1,s}(\theta_1)\tau^*_{s+K}(\theta_2)\\
&= \left(\sum_{j=0}^{\infty}\sum_{k=-s}^{\infty} - \sum_{j=0}^{\infty}\sum_{k=-s}^{K} - \sum_{j=0}^{J-m-s-1}\sum_{k=-s}^{\infty} + \sum_{j=0}^{J-m-s-1}\sum_{k=-s}^{K}\right)\pi_{J,K}\rho_1^{j+m+s+1-J}\rho_2^{k-K}\tilde{Y}^*_{j+m,s}(\theta_1)\tau^*_{s+k}(\theta_2)\\
&\quad + \left(\sum_{j=0}^{J-m-s-1}\sum_{k=-s}^{\infty} - \sum_{j=0}^{J-m-s-1}\sum_{k=-s}^{K-1}\right)\sum_{i=1}^{J+1}c_{1,i}e_{1,i;j+m+s+1}x_{1,i}^{k}\tilde{Y}^*_{j+m,s}(\theta_1)\tau^*_{s+k}(\theta_2)\\
&\quad + \left(\sum_{j=0}^{\infty}\sum_{k=0}^{K} - \sum_{j=0}^{J-m-s-2}\sum_{k=0}^{K}\right)\sum_{i=1}^{K+1}c_{2,i}e_{2,i;k}x_{2,i}^{j+m+s+1}\tilde{Y}^*_{j+m,s}(\theta_1)\tau^*_{s+k}(\theta_2)\\
&\quad + \sum_{j=0}^{J-m-s-2}\sum_{k=0}^{K-1}\pi_{j+m+s+1,k}\tilde{Y}^*_{j+m,s}(\theta_1)\tau^*_{s+k}(\theta_2) - \pi_{J,K}\tilde{Y}^*_{J-s-1,s}(\theta_1)\tau^*_{s+K}(\theta_2).
\end{aligned}$$

Changing the summation variable k in the sums starting from $k=-s$, we get

$$\begin{aligned}
\eta_{ms}^{-1}\tilde{T}^*_{ms}(\theta_1,\theta_2) &= \pi_{J,K}\rho_1^{m+s+1-J}\rho_2^{-K-s}\\
&\quad\times\left(F_{ms}(\rho_1,\rho_2;\theta_1,\theta_2) - \sum_{k=0}^{K+s}\frac{F_{ms}^{(0,k)}(\rho_1,0;\theta_1,\theta_2)\rho_2^k}{k!} - \sum_{j=0}^{J-m-s-1}\frac{F_{ms}^{(j,0)}(0,\rho_2;\theta_1,\theta_2)\rho_1^j}{j!}\right)\\
&\quad + \pi_{J,K}\rho_1^{m+s+1-J}\rho_2^{-K-s}\sum_{j=0}^{J-m-s-1}\sum_{k=0}^{K+s}\rho_1^j\rho_2^k\frac{F_{ms}^{(j,k)}(0,0,\theta_1,\theta_2)}{j!k!}\\
&\quad + \sum_{i=1}^{J+1}x_{1,i}^{-s}\sum_{j=0}^{J-m-s-1}c_{1,i}e_{1,i;j+m+s+1}\frac{F_{ms}^{(j,0)}(0,x_{1,i};\theta_1,\theta_2)}{j!}\\
&\quad - \sum_{i=1}^{J+1}x_{1,i}^{-s}\sum_{j=0}^{J-m-s-1}\sum_{k=0}^{K+s-1}c_{1,i}e_{1,i;j+m+s+1}x_{1,i}^{k}\frac{F_{ms}^{(j,k)}(0,0,\theta_1,\theta_2)}{j!k!}
\end{aligned}$$

[13] This quantity is also equal to $\sum_{i=1}^{K+1}c_{2,i}\vec{e}_{2,i;K}x_{2,i}^{J}$ in an exact model but here the closeness of the values of the two expressions is used as a stopping condition in the algorithm for computing equilibrium state probabilities [17].

$$+\sum_{i=1}^{K+1} x_{2,i}^{m+s+1} \sum_{k=0}^{K} c_{2,i} e_{2,i;k} \frac{F_{ms}^{(0,s+k)}(x_{2,i},0;\theta_1,\theta_2)}{(s+k)!}$$
$$-\sum_{i=1}^{K+1}\sum_{k=0}^{K} x_{2,i}^{m+s+1} \sum_{j=0}^{J-m-s-2} c_{2,i} e_{2,i;k} x_{2,i}^{j} \frac{F_{ms}^{(j,s+k)}(0,0,\theta_1,\theta_2)}{j!(s+k)!}$$
$$+\sum_{j=0}^{J-m-s-2}\sum_{k=0}^{K-1} \pi_{j+m+s+1,k} \frac{F_{ms}^{(j,s+k)}(0,0,\theta_1,\theta_2)}{j!(s+k)!}$$
$$-\pi_{J,K} \frac{F_{ms}^{(J-m-s-1,s+K)}(0,0,\theta_1,\theta_2)}{(J-m-s-1)!(s+K)!},$$

as required. □

H DEPENDENCIES AMONG THEOREMS, PROPOSITIONS, AND LEMMAS

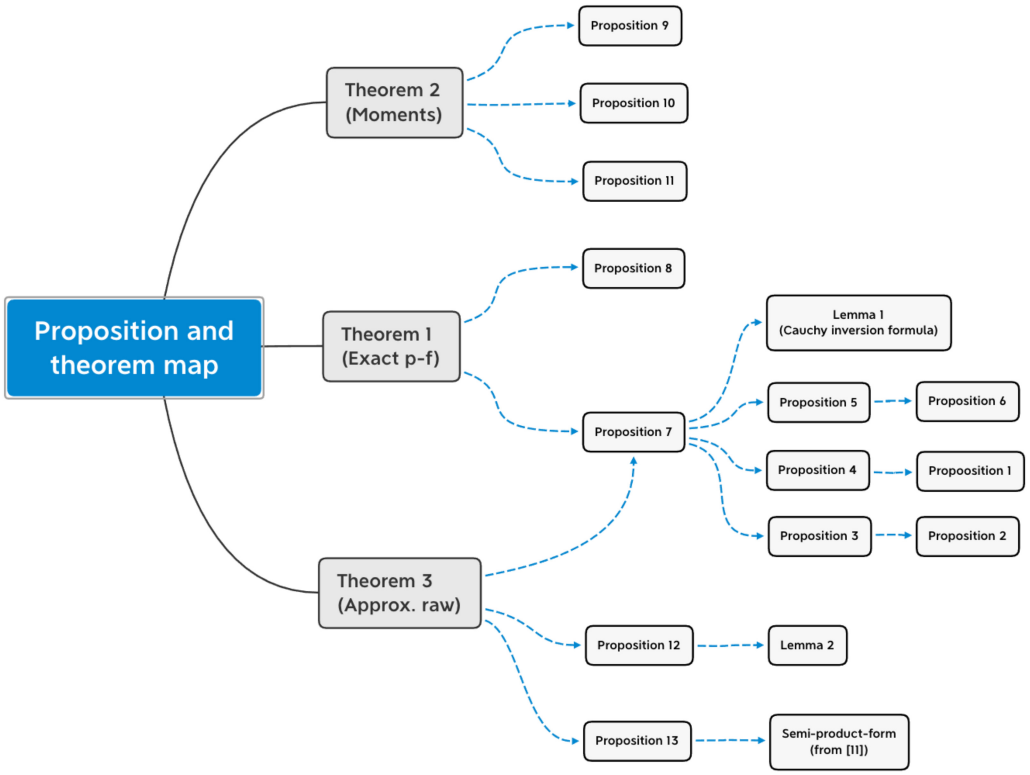

Fig. 11. Roadmap of dependencies among main results.

REFERENCES

[1] O. J. Boxma, F. P. Kelly, and A. G. Konheim. 1984. The product form for sojourn time distributions in cyclic exponential queues. *Journal of the ACM* 31, 1 (Jan. 1984), 128–133. DOI: https://doi.org/10.1145/2422.322419
[2] X. Chao, M. Miyazawa, and M. Pinedo. 1999. *Queueing Networks: Customers, Signals and Product form Solutions*. Wiley.
[3] Xi Chen, Chin Pang Ho, Rasha Osman, Peter G. Harrison, and William J. Knottenbelt. 2014. Understanding, modelling, and improving the performance of web applications in multicore virtualised environments. In *Proceedings of the 5th*

ACM/SPEC International Conference on Performance Engineering (ICPE'14). ACM, New York, NY, 197–207. DOI: https://doi.org/10.1145/2568088.2568102

[4] Chiahon Chien. 1995. Batch size selection for the batch means method. In *Proceedings of the 1994 Winter Simulation Conference*, J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila (Eds.). 345–352. DOI: https://doi.org/10.1109/WSC.1994.717192

[5] We-Min Chow. 1980. The cycle time distribution of exponential cyclic queues. *Journal of the ACM* 27, 2 (April 1980), 281–286. DOI: https://doi.org/10.1145/322186.322193

[6] Edward G. Coffman Jr., Guy Fayolle, and Isi Mitrani. 1988. Two queues with alternating service periods. In *Proceedings of the 12th IFIP WG 7.3 International Symposium on Computer Performance Modelling, Measurement and Evaluation (Performance'87)*. North-Holland Publishing Co., Amsterdam, The Netherlands, 227–239. http://dl.acm.org/citation.cfm?id=647412.724892

[7] Hans Daduna. 1982. Passage times for overtake-free paths in Gordon-Newell networks. *Advances in Applied Probability* 14, 3 (1982), 672–686. DOI: https://doi.org/10.2307/1426680

[8] Paul D. Ezhilchelvan, Isi Mitrani, and Jim Webber. 2018. On the degradation of distributed graph databases with eventual consistency. In *Computer Performance Engineering - 15th European Workshop (EPEW'18), Proceedings*. 1–13. DOI: https://doi.org/10.1007/978-3-030-02227-3_1

[9] Peter W. Glynn. 2006. Simulation algorithms for regenerative processes. In *Simulation*, Shane G. Henderson and Barry L. Nelson (Eds.). Handbooks in Operations Research and Management Science, Vol. 13. Elsevier, 477–500. DOI: https://doi.org/10.1016/S0927-0507(06)13016-9

[10] D. Gross and C. M. Harris. 1985. *Fundamentals of Queueing Theory*. Wiley-Sons.

[11] Jim Handy. 2013. *How Controllers Maximize SSD Life*. Technical Report. SNIA. Retrieved from https://www.snia.org/sites/default/files/SSSITECHNOTES_HowControllersMaximizeSSDLife.pdf.

[12] P. G. Harrison. 1984. The distribution of cycle times in tree-like networks of queues. *Computer Journal* 27, 1 (1984), 27–36.

[13] P. G. Harrison. 2003. Turning back time in Markovian process algebra. *Theoretical Computer Science* 290, 3 (2003), 1947–1986. http://aesop.doc.ic.ac.uk/pubs/rcat/

[14] P. G. Harrison. 2018. Product-form queueing networks with batches. In *European Workshop on Performance Engineering*. Springer, 250–264.

[15] P. G. Harrison. 2020. A semi-product-form for a pair of queues with finite batches: Equilibrium state probabilities and response time densities. *Performance Evaluation* 143 (2020). DOI: https://doi.org/10.1016/j.peva.2020.102120

[16] P. G. Harrison and Naresh M. Patel. 1992. *Performance Modelling of Communication Networks and Computer Architectures*. Addison-Wesley. http://aesop.doc.ic.ac.uk/pubs/perf-mod-com-net/

[17] P. G. Harrison. 2019. A semi-product-form for the equilibrium state probabilities in a pair of queues with finite batches. In *Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'19)*. ACM, New York, NY, 31–38. DOI: https://doi.org/10.1145/3306309.3306316

[18] Peter G. Harrison, Richard A. Hayden, and William J. Knottenbelt. 2013. Product-forms in batch networks: Approximation and asymptotics. *Performance Evaluation* 70, 10 (Oct. 2013), 822–840. DOI: https://doi.org/10.1016/j.peva.2013.08.011

[19] W. Henderson and P. Taylor. 1990. Product form in networks of queues with batch arrivals and batch services. *Queueing Systems* 6 (1990), 71–88.

[20] Donald L. Iglehart and Ward Whitt. 1970. Multiple channel queues in heavy traffic. I. *Advances in Applied Probability* 2, 1 (1970), 150–177. http://www.jstor.org/stable/3518347

[21] Donald L. Iglehart and Ward Whitt. 1970. Multiple channel queues in heavy traffic. II: Sequences, networks, and batches. *Advances in Applied Probability* 2, 2 (1970), 355–369. http://www.jstor.org/stable/1426324

[22] E. G. Coffman Jr., G. Fayolle, and I. Mitrani. 1986. Sojourn times in a tandem queue with overtaking: Reduction to a boundary value problem. *Communications in Statistics. Stochastic Models* 2, 1 (1986), 43–65. DOI: https://doi.org/10.1080/15326348608807024

[23] O. Kella and W. Whitt. 1992. A tandem fluid network with Lévy input. In *Queueing and Related Models*, U.N. Bhat and I. V. Basawa (Eds.). Oxford University Press, 112–128.

[24] Offer Kella and Ward Whitt. 1996. Stability and structural properties of stochastic storage networks. *Journal of Applied Probability* 33, 4 (1996), 1169–1180. http://www.jstor.org/stable/3214994

[25] F. P. Kelly. 1979. *Reversibility and Stochastic Networks*. Wiley.

[26] F. P. Kelly and P. K. Pollett. 1983. Sojourn times in closed queueing networks. *Advances in Applied Probability* 15 (1983), 638–656.

[27] Alexander Klemm, Christoph Lindemann, and Marco Lohmann. 2002. *Traffic Modeling of IP Networks Using the Batch Markovian Arrival Process*. Springer, Berlin, 92–110. DOI: https://doi.org/10.1007/3-540-46029-2_6

[28] A. S. Lebrecht, N. J. Dingle, P. G. Harrison, W. J. Knottenbelt, and S. Zertal. 2009. Using bulk arrivals to model I/O request response time distributions in zoned disks and RAID systems. In *Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'09)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Article 23, 10 pages. DOI: https://doi.org/10.4108/ICST.VALUETOOLS2009.7787

[29] Mihaela Mitici, Jasper Goseling, Jan-Kees van Ommeren, Maurits de Graaf, and Richard J. Boucherie. 2017. On a tandem queue with batch service and its applications in wireless sensor networks. *Queueing Systems* 87, 1–2 (Jun 2017), 81–93. DOI: https://doi.org/10.1007/s11134-017-9534-1

[30] I. Mitrani. 1985. Response time problems in communication networks. *Journal of the Royal Statistical Society. Series B (Methodological)* 47, 3 (1985), 396–406. http://www.jstor.org/stable/2345774

[31] Isi Mitrani. 2013. Managing performance and power consumption in a server farm. *Annals of Operations Research* 202, 1 (Jan. 2013), 121–134. DOI: https://doi.org/10.1007/s10479-011-0932-1

[32] Isi Mitrani. 2018. Multi-class resource sharing with preemptive priorities. *Probability in the Engineering and Informational Sciences* 32, 3 (2018), 323–339. DOI: https://doi.org/10.1017/S0269964817000286

[33] I. Mitrani and R. Chakka. 1995. Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method. *Performance Evaluation* 23 (1995), 241–260.

[34] Randolph Nelson. 2010. *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling* (1st ed.). Springer Publishing Company, Incorporated.

[35] A. E. Papathanasiou and M. L. Scott. 2003. Energy efficiency through burstiness. In *5th IEEE Workshop on Mobile Computing Systems and Applications*.

[36] Fernando C. Pereira and Touradj Ebrahimi. 2002. *The MPEG-4 Book*. Prentice Hall PTR, .

[37] C. H. Sauer and K. M. Chandy. 1975. Approximate analysis of central server models. *IBM Journal of Research and Development* 19, 3 (May 1975), 301–313. DOI: https://doi.org/10.1147/rd.193.0301

[38] R. Schassberger and H. Daduna. 1983. The time for a round trip in a cycle of exponential queues. *Journal of the ACM* 30, 1 (Jan. 1983), 146–150. DOI: https://doi.org/10.1145/322358.322369

[39] E. A. van Doorn and J. K. Regterschot. 1988. Conditional PASTA. *Operations Research Letters* 7 (1988), 229–232.

[40] J. Walrand and P. Varaiya. 1980. Sojourn times and the overtaking condition in Jacksonian networks. *Advances in Applied Probability* 12, 4 (1980), 1000–1018. DOI: https://doi.org/10.2307/1426753

[41] Ronald W. Wolff. 1982. Poisson arrivals see time averages. *Operations Research* 30, 2 (1982), 223–231. http://www.jstor.org/stable/170165

Received June 2019; revised January 2021; accepted February 2021